



Job-work fit as a determinant of the acceptance of large-scale agile methodology

Dinesh Batra

IS and Business Analytics, Florida International University, United States

ARTICLE INFO

Article history:

Received 19 December 2018

Revised 18 February 2020

Accepted 17 March 2020

Available online 20 May 2020

Keywords:

Large-scale agile methodology

Perceived usefulness

Compatibility

Subjective norm

Mandatoriness

External support

ABSTRACT

The declaration of the Agile Manifesto in 2001 was hailed as a paradigmatic shift in the development of software systems. Initially, agile development was typically deployed in projects with a single team with fewer than ten members. Recently, a new kind of agile development has emerged for larger projects that may be referred to as large-scale agile methodology (LSAM). The taxonomy of LSAM is more enhanced and subsumes the Agile Manifesto. Based on the software development literature, the study investigates five important antecedents of methodology acceptance: perceived usefulness, compatibility, subjective norm, mandatoriness, and external support. As a result of the initial PLS analysis, the study introduces the notion of Job-Work Fit as a second-order construct composed of perceived usefulness and compatibility. Based on a survey of 123 respondents, the study finds that the construct job-work fit significantly explains the LSAM acceptance for large projects. Job-work fit also mediates the relationships between subjective norm and external support with LSAM acceptance. This research suggests that as the stakes become higher because of project size and agility, job-work fit emerges as the central determinant of LSAM acceptance.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

The declaration of the Agile Manifesto in 2001 (Fowler and Highsmith, 2001) signaled a paradigmatic shift in software development. In contrast to the process-heavy methods that had difficulty in accommodating requirement changes, agile methods emphasized individuals, interactions, and customer value while welcoming requirement changes. However, during the initial years, agile methods were generally employed in small projects (Dyba and Dingsøyr, 2008). For example, Extreme Programming (XP) is considered as not suitable for projects that had more than ten members (Beck, 2000). As agile development became more commonplace, both researchers and practitioners called for an enhanced methodology that was based on agile concepts but could be applied to larger projects (Leffingwell, 2007). This call led to large-scale agile methodology (LSAM), which has manifested in new methods such as Scaled Agile Framework (SAFe) (Knaster and Leffingwell, 2017) and Large-Scale Scrum (LeSS) (Larman and Vodde, 2016). A large project can be defined as having at least 2–9 teams, with each team having from five to nine members (Dingsøyr et al., 2014). As recommended by (Hannay and Benes-

tad, 2010), this study operationalized a large size project as one that had 20 or more engaged members.

Currently, there is no consensus on how a large project should be developed. It can be delivered using waterfall, agile methods like Scrum for large projects, or hybrid methods (Theocharis et al., 2015; VersionOne, 2018). LSAM attempts to reconcile principles from different approaches and, despite concerns such as coordination difficulties, has a distinct likelihood of becoming the preferred approach (Dikert et al., 2016; Rolland et al., 2016). Over time, LSAM has gained market acceptance (Ebert and Paasivaara, 2017; Paasivaara et al., 2012; Van Waardenburg and Van Vliet, 2013) and according to the 2018 VersionOne and recent surveys (Alqudah and Razali, 2016; Dingsøyr et al., 2019), LSAM is gaining adoption by using the following methods: SAFe, LeSS, Scrum of Scrums, hybrid, and agile hybrid, along with the popular agile methods such as Scrum and Kanban.

However, the mere acquisition of technology does not imply that it is assimilated or accepted (Downs and Mohr, 1976; Fichman and Kemerer, 1997; Tornatzky and Klein, 1982). The literature does not provide a taxonomy of “large-scale agile,” which gets usually reduced to the use of a named LSAM like SAFe or LeSS. Thus, an initial research step is to operationalize the term large-scale agile methodology. Conboy (2009) improves the notion of “agile” and provides a taxonomy of “agility” that is more theoretical and comprehensive. In this paper, the taxonomy proposed

E-mail address: batra@fiu.edu

by Conboy (2009) is extended to offer an operationalization of the large-scale agile construct. The resulting definition is then used to develop measures of the LSAM construct.

The next research objective is to identify the antecedents of software methodology adoption. The research on adoption of structured methods has identified precursors such as perceived usefulness/relative advantage/job-fit, compatibility, subjective norm, and mandatoriness (Hardgrave et al., 2003; Riemenschneider et al., 2002) and external support (Roberts et al., 1998, 2001). In slight contrast, the research on the adoption of agile methods (Vijayasathya and Turk, 2012) has identified factors such as subjective norm and training, but not perceived usefulness and compatibility. An earlier meta-analysis of antecedents in the adoption literature (Tornatzky and Klein, 1982) revealed only two factors – relative advantage and compatibility – as significant. Using these streams of related works, the research reported in this paper builds a model for identifying the crucial antecedents of LSAM acceptance. The Smart PLS technique (Hair et al., 2016) is used to analyze the research model.

The research delved into the essential research question: which factors determine the acceptance and use of large-scale agile methodology (LSAM)? Based on the literature, the study considered the following constructs: perceived usefulness, compatibility, subjective norm, mandatoriness, and external support as the independent variables and acceptance of LSAM as the dependent variable. Experience with agile development and the size of the organization were considered as single indicator variables for additional analysis. Data were collected from 123 participants. The paper describes the operationalization of a large project, related work, measuring the constructs, methodology, initial and final models, results, discussion, limitations, and conclusions. Appendix A includes the questionnaire.

The study makes five main contributions. First, the large-scale agile construct is operationalized. Second, the job-work fit is proposed as a higher-order construct composed of perceived usefulness and compatibility. Third, the job-work fit construct has a dominant effect on LSAM acceptance. Fourth, the job-work fit showed significant results as a mediating variable for the subjective norm and external support. Finally, the study indicates that mandatoriness, which has been shown to have significant effects on structured methodology acceptance, does not have significant effects on the LSAM acceptance.

2. Operationalization of large-scale agile methodology

A systems or software development methodology (SDM) can be defined as a documented collection of policies, processes, and procedures to improve the software development process regarding the increased productivity of information technology (IT) personnel and higher quality of the final IT solutions (Chan and Thong, 2009). A class of SDMs is based on an underlying manifesto, a set of vital aspects, or a set of principles. Nerur et al. (2005) provide twelve distinguishing characteristics of structured and agile methodologies. Underlying values and goals are better indicators of the essence of a method (Conboy, 2009). Operationalization of a methodology can specify the underlying values and address the illusory acceptance problem, which is the gap between adoption and assimilation (Downs and Mohr, 1976; Fichman and Kemerer, 1997, 1999). Boehm and Turner (2004, p. 6) provide an example of agile misuse: a vendor claimed that his organization was using XP, but when asked specific practices such as pair programming, refactoring, and game planning, he said that all these practices were not adopted. When asked, “Then what are you doing?” the answer was, “We don’t document anything!”. Similarly, inadequate customer involvement can mitigate the effectiveness of the use of an agile method (Hoda et al., 2011).

2.1. Operationalization of LSAM based on agile and large-agile literature

After the declaration of the Agile Manifesto (Fowler and Highsmith, 2001), agile methods such as XP (Beck, 2000) and Scrum (Schwaber, 2004) became popular. Each particular method had a specific set of practices that supplemented the values specified in the Agile Manifesto. Many researchers such as (Boehm and Turner, 2004; Cao et al., 2009; Conboy, 2009; Jacobson and Stimson, 2018) note that agile methods are appropriated to suit the context. In an attempt to improve the notion of “agility” in agile development, Conboy (2009) evaluated related streams of literature and came up with the following definition: “the continual readiness of an ISD method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment.” This more comprehensive definition addressed not only the agile values (e.g., embrace change) but also precepts such as lean (i.e., customer value), and implicitly referred to continual improvement (i.e., learn from change) and systems approach (i.e., collective components and relationships).

In the case of agile development, the four fundamental values are part of the manifesto; but for large agile, there is no explicit manifesto, and researchers qualify LSAM by using names of popular methodologies. Despite the lack of a clear LSAM framework, the taxonomy proposed by Conboy (2009) seems to have been implicitly embraced and extended by the LSAM literature. Specifically, literature based on the most widely used methods list similar precepts and have additional recommendations to address challenges such as coordination (Bick et al., 2018) and self-organization (Dikert et al., 2016). SAFe (Knaster and Leffingwell, 2017) and LeSS (Larman and Vodde, 2016) are popular LSAM methodologies. Knaster and Leffingwell (2017, pp. 6–8) discuss three critical bodies of knowledge – agile development, systems thinking, and lean product development – and advocate continuous improvement as part of embracing “kaizen” (p. 8), and transparency for building trust, which is essential for performance, innovation, risk-taking, and relentless improvement (p. 13). Larman and Vodde (2016, pp. 10–12) list the following principles: Large-scale Scrum is Scrum, transparency, more with less, whole-product focus, customer-centric, continuous improvement towards perfection, lean thinking, systems thinking, empirical-process control, and queuing theory.

The four facets stipulated by Conboy (2009) – agile values, lean thinking, and continuous improvement – are also addressed by SAFe and LeSS. The systems approach, implied by Conboy (2009) and discussed in textbooks, is explicitly included by SAFe and LeSS. This research added two more – transparency and decentralized decision making – from the two books and research literature. Lack of coordination is a prominent barrier in agile and LSAM projects (Bick et al., 2018; Bjørnson et al., 2018; Østdahl, 2017; Scheerer et al., 2014; Strode et al., 2012). Transparency is vital in a large project for eliminating fear, minimizing coordination and alignment costs, and enhancing trust (Knaster and Leffingwell, 2017). Self-organization is crucial for achieving agility (Hoda et al., 2012, 2013; Lee and Xia, 2010; Moe et al., 2008), but because self-organization may be more difficult in LSAM, Knaster and Leffingwell (2017) propose decentralized decision-making as a more feasible principle.

Building on the (Conboy, 2009) taxonomy, this paper defines large-scale agile as a methodology for achieving agility in a project described as large by empirical standards and contributing to perceived customer value by facilitating effective, synchronized development of its collective components and relationships through continuous improvement, decentralized decision-making

and transparency. In summary, this research considered six critical measures of LSAM acceptance: agile values and principles, systems thinking, lean product development, transparency, continual improvement, and decentralized decision making. Although each literature source may list some additional facets, the six measures are adequate in serving as a common denominator of LSAM projects and can, thus, be used to measure the extent of its acceptance.

2.2. Measure of “Large” in large-scale agile

This research does not evaluate the potential adoption of LSAM; however, it evaluates the acceptance of LSAM through the six characteristics assuming that an individual has participated in a large agile project. Thus, it was essential to operationalize a large project. “Large” is a qualitative attribute and, when measured quantitatively, subject to some interpretation. The literature has employed the number of engaged members as well as the number of teams as the primary measures. Because the number of members can vary from five to nine (Dingsøyr et al., 2019) or three to nine (Knaster and Leffingwell, 2017), the researcher operationalized a large-size project as one that has 20 or more members, which is consistent with the (Hannay and Benestad, 2010) recommendation. Because “large” is always qualified in a context, one way to make the measure more objective is through the use of percentiles.

For any domain, including software development, the 80th percentile may be an acceptable operationalization for a large amount or quantity. Two studies have provided enough statistical information that may offer a reasonable measure of the number of engaged members in a large project. In a study of IT projects, El Emam and Koru (2008) provided a chart showing the project distribution, which suggested that about 80% of projects had 20 or fewer and about 90% of the projects had 50 or fewer technical members. We can assume that the number of engaged members may be slightly larger than the number of technical members because of customer roles like Product Owner and internal roles like the project manager. The (Pendharkar and Rodger, 2009) study provided statistical data but did not show a histogram. In their survey of the relationship between software development team size and cost, the data selected for final analysis had a mean of 8.13, a standard deviation of 9.05, and a range of 1–77 persons (although the initial data set had a similar mean but greater standard deviation because of outliers). Thus, it is reasonable to assume that about 75–80% of the project sizes would be below 20.

Dingsøyr et al. (2019) provide several frameworks with respective project sizes; for example, LeSS is suitable for two to seven teams, DAD for one to many, and Nexus for three to nine teams, where each team can have from five to nine members. They assert that an LSAM project should have at least two teams. If the minimum number of LSAM teams is two to three and the number of members per team is from five to nine, then a large project can have a minimum range from 10 to 18 members. Based on the evaluation of project distributions from the two empirical studies, the lower end (i.e., 10) of the range seems questionable. SFAE can have five to twelve agile teams, where each team has from three to nine members (Knaster and Leffingwell, 2017); consequently, the lower minimum is 15, which is somewhat lesser than what is suggested by empirical data. Thus, a minimum of 20 members, as indicated by (Hannay and Benestad, 2010), seems a viable measure of a large agile project.

3. Related work

Based on a meta-analysis of seventy-five articles, Tornatzky and Klein (1982) concluded that three innovation characteristics – compatibility, relative advantage, and complexity – had the most con-

sistent relationships to innovation adoption. Technically, complexity is not statistically significant because the $p = 0.062$ was higher than the 0.05 cutoff, but they deemed it reasonably close. Complexity is an inverse measure of perceived ease-of-use; however, perceived ease-of-use was found insignificant in the theoretical models employed in the (Riemenschneider et al., 2002) and most other methodology acceptance studies. The current study, therefore, did not consider perceived ease-of-use.

Relative advantage and compatibility are likely to participate as essential variables in LSAM acceptance. Relative advantage is equivalent to perceived usefulness and job-fit, while compatibility can be regarded as equivalent to work-fit. Because relative advantage and compatibility can be highly correlated (Moore and Benbasat, 1991), the statistical treatment of the two constructs may necessitate the design of a higher-order construct. The operationalizations of relative advantage and compatibility depend on the domain. For example, on relative advantage, (Rogers, 1983, p. 229) notes: “The nature of the innovation determines what type of relative advantage (economic, social, and the like) is important to adopters.”

A literature survey of the acceptance of software methodologies (Chan and Thong, 2009) indicates limited theory-based empirical reviews on agile development. However, Vijayasarathy and Turk (2012) provide a comprehensive, theory-based agile acceptance study. They drew on intention-based theories and found that subjective norm was the primary factor in influencing software developers’ use of agile methods. Perceived benefit emerged as a significant predictor of agile methodology use only if adopters faced hindrances to their agile practices. Furthermore, they found training as a positive predictor and software development experience and organization size as negative predictors of agile development use. They did not study the effects of compatibility. They measured agile development acceptance not by recording adoption or non-adoption but on the extent of use of four methods or practices: Extreme Programming, agile unified process, pair programming, and test-first development (although one could argue that pair programming and test-first development are already part of Extreme Programming). Thus, rather than simple yes-no adoption, their study focused on the extent of implementation, an approach that was also followed in the study reported in this paper.

To assess the acceptance of structured methodologies, Riemenschneider et al. (2002) conducted a comprehensive study on acceptance of structured methods in an organization and tested five theories – Technology Acceptance Model (TAM) (Davis et al., 1989), TAM2 (Venkatesh and Davis, 2000), Perceived Characteristics of Innovating (Moore and Benbasat, 1991; Rogers, 1983), Theory of Planned Behavior (Ajzen, 1991), and Model of Personal Computer Utilization (Thompson et al., 1991). The survey included a comprehensive list of constructs: usefulness/relative advantage/attitude/job fit, ease-of-use/complexity, subjective norm/social factors, voluntariness/mandatoriness, compatibility, affect, result demonstrability, image, visibility, perceived behavioral control – internal, perceived behavioral control – external, facilitating conditions, and career consequences. Note that the study considered some constructs (e.g., perceived usefulness, relative advantage, attitude, and job fit) from different theories as equivalent. A survey of organizational culture (Iivari and Huisman, 2007) hypothesized mandatoriness as a significant predictor of SDM deployment. They found that the deployment of methodologies by IS developers is primarily associated with a hierarchical culture oriented toward security, order, and routinization; interestingly, IT managers were more critical of the deployment of methodologies. Four constructs emerged as significant: perceived usefulness, subjective norm, mandatoriness, and compatibility.

The same four constructs were also found significant in other studies on structured software development. [Hardgrave et al. \(2003\)](#) found that perceived usefulness, social pressure, compatibility, and mandatoriness predicted intention to follow the structured development methodology. Perceived complexity did not emerge as an important antecedent. They determined that this pattern of intention determinants is different from that typically observed in studies of information technology tool adoption. [Kacmar et al. \(2009\)](#) found that perceived usefulness had a significant effect on the fit of methodology with client problems and the efficacy of methods used.

[Templeton and Byrd \(2003\)](#) surveyed 47 subjects to examine the constructs from the Perceived Characteristics of Innovating ([Moore and Benbasat, 1991](#)) and TAM ([Davis et al., 1989](#)) models to predict relative advantage. Both perceived ease-of-use (0.781) and compatibility (0.727) were strongly correlated to relative advantage. As is widely accepted in TAM and TAM2 models, perceived ease-of-use emerged as the significant determinant of relative advantage. Both perceived ease-of-use and trialability were substantial determinants of compatibility. Their study did not include a measure of adoption or acceptance of software methodology.

External support has been found significant in many studies. [Roberts et al. \(2001\)](#) found that external support helped enhance knowledge development and the implementation of an SDM. [Roberts et al. \(1998\)](#) found five factors that were important to companies implementing SDMs: organizational SDM transition, functional management involvement/support, SDM transition, use of models, and external support. According to [Roberts et al. \(1998\)](#) study, external support subsumes training but also includes support such as the use of external consultants. [Khalifa and Verner \(2000\)](#) found two constructs, facilitating conditions and process quality, as drivers of the method used. The term, facilitating conditions, includes attributes such as organizational support for adopting emerging technologies. In the current study, the construct external support includes items such as specialized instruction, coaching resource, a network of support, and other help and resources (e.g., any kind of training).

Some studies have focused on the acceptance of object-oriented technology. The object-oriented technology is strictly not a methodology because it does not have prescribed processes, although it is associated with techniques and languages such as Unified Modeling Language (UML). [Hardgrave and Johnson \(2003\)](#) found that usefulness mediated the effects of the subjective norm and perceived behavioral control. [Johnson \(1999\)](#) established the relevance of the TAM model to software development methodology and developed measures for perceived usefulness and ease-of-use for object-oriented analysis and design (OOAD) technology.

There is some discrepancy in the acceptance factors for the agile, which emphasizes subjective norm and training, and the structured approach, which underlines perceived usefulness, compatibility, subjective norm, and mandatoriness. It is not clear which factors are essential when the agile project scales up. The LSAM is an extension of agile methodology but deals with larger projects, and with the visible higher stakes, it is improbable that participants would adopt and accept it if they do not deem it as useful. After all, project risk about factors such as cost, delivery time, software quality, and customer satisfaction, which are possible dimensions for assessing perceived usefulness, is ascribed to both large size and changing requirements ([Barki et al., 1993](#)). Perceived usefulness/relative advantage is the critical factor in the ([Riemenschneider et al., 2002](#)) and many other studies on structured development ([Chan and Thong, 2009](#)). Consequently, this factor should be scrutinized more closely when agile development scales up to LSAM.

4. Measuring the constructs

The research reported in this paper was based on a collection of the constructs commonly found significant in the theory-based acceptance literature: perceived usefulness, compatibility, subjective norm, mandatoriness, and external support. The current study adapted the independent variables based on ([Vijayasathy and Turk, 2012](#)) and ([Riemenschneider et al., 2002](#)) studies as well as the popular literature such as [Knaster and Leffingwell \(2017\)](#) and [Larman and Vodde \(2016\)](#). The measurement of LSAM acceptance and compatibility presented special adaptation issues. [Appendix A](#) shows the indicators of the constructs. The data and the analysis procedure are available at <https://data.mendeley.com/datasets/227b5x4vg6/2>.

4.1. LSAM acceptance

[Tornatzky and Klein \(1982\)](#) recommend that for the more complex (socio-technical) innovations, the dominant characteristics of the innovation should be used to measure the acceptance. As detailed in [Section 2.1](#), this research considered six critical measures of LSAM acceptance: agile values and principles, systems thinking, lean product development, transparency, continual improvement, and decentralized decision making.

4.2. Perceived usefulness and compatibility

Relative advantage is the degree to which an innovation is perceived as being better than its precursor ([Rogers, 1983](#)) while perceived usefulness is the degree to which an individual believes that using a particular system would enhance his or her performance ([Davis, 1989](#)), although [Moore and Benbasat \(1991\)](#) view that the notion of relative is implicit in perceived usefulness. Studies involving perceived usefulness may not always measure particular aspects of the domain; however, software development usually targets well-known outcomes. [Vijayasathy and Turk \(2012\)](#) had measures for the construct perceived benefits relating to quality, delivery time, development costs, customer needs, reusable code, and flexibility in development. The current study used a similar approach for measuring perceived usefulness.

Compatibility is defined as the degree to which an innovation is perceived as being consistent with the existing values, needs, and past experiences of potential adopters ([Rogers, 1983](#)). In software development literature, compatibility has aligned with the “compatibility of the work aspect” of developing a system. For example, [Riemenschneider et al. \(2002\)](#) used three measures “ADM is compatible with the way I develop systems,” “Using ADM is compatible with all aspects of my work,” and “Using ADM fits well with the way I work.” This study employed four items for measuring compatibility. The first item is generic, but the remaining items are specific and try to capture the nature of LSAM work. The second and the third items capture two distinguishing characteristics of LSAM: that it is used for large projects and that it should respond to changes in requirements. The fourth item - reconciling different views of stakeholders - stems from the ([Bick et al., 2018](#); [Dingsøyr et al., 2018](#); [Rolland et al., 2016](#)) studies, which outline planning misalignment and lack of dependency awareness among stakeholders as the pressing inter-team challenges in LSAM. Thus, the last three items are domain-specific and qualify what ([Riemenschneider et al., 2002](#)) mentioned generically as “compatible with all aspects of work” and “with the way I work.” As shown later, the adapted construct exhibited reliability and convergent validity.

Table 1
Project size, organization size, and agile experience demographics.

Project size		Organization size		Agile experience	
Size range	count by percent	Size range	Count by percent	Experience in years	Count by percent
20–24	29.3	<40	17.1	1–4	16.3
25–30	26	41–200	18.7	5–6	30.1
31–50	28.5	201–500	17.9	7–9	19.5
51–100	10.6	501–1000	16.3	10–12	21.1
101–500	4.1	1001–4000	20.3	13–20	9.8
500–750	1.6	>4000	9.8	21–25	3.3

4.3. Subjective norm, external support, and mandatoriness

Subjective norm is the extent to which important referent groups/individuals/organizations influence adoption and use (Vijayasathy and Turk, 2012). External support is the use of training and consultants (Roberts et al., 1998). Mandatoriness captures the extent to which the desired behavior is made mandatory (Iivari and Huisman, 2007). The measures for these three constructs were adapted from (Riemenschneider et al., 2002; Roberts et al., 1998; Vijayasathy and Turk, 2012). Appendix A provides the items for all the constructs, which were modified to fit with the LSAM domain. As explained in the next section, all constructs passed the requirements of reliability and convergent validity, while two constructs were combined into a higher-order construct to resolve discriminant validity.

5. Methodology

The research model was evaluated based on the partial least squares for a structural equation modeling (PLS-SEM) approach (Richter et al., 2015) using the SmartPLS version 3 software (Ringle et al., 2015). The PLS-SEM approach was selected because the purpose of the study was exploratory rather than confirmatory. PLS-SEM is preferred when conducting exploratory research (Gefen et al., 2000; Vinzi et al., 2010). The research objective was prediction rather than the confirmation of structural relationships (Hair et al., 2011). It was essential to determine the explained variance (R-square) of the dependent variables and the strength of the relationships (Hair et al., 2012). The use of the alternative, covariance-based SEM (CB-SEM) approach is appropriate for confirmatory research (Diamantopoulos and Siguaw, 2000; Hair et al., 2011).

The study required that the survey respondent had engaged in a large, agile project, which was operationalized as an agile project that involved 20 or more participants. The researcher had attended many local meetups on agile development for more than a year and found that although agile methods are commonly used, large-size agile projects are rare. Participants indicated that they rarely conduct projects that have more than 35 members. Such encounters and discussions, however anecdotal, suggested that it would not be easy to collect data from a limited set of local companies. The goal of the researcher was to collect data from a substantial number of large-size agile projects from different contexts. Collecting data from a single or few companies would have restricted the possible diversity of opinions and introduced bias in measurement. Thus, the researcher sought the help of data collection companies.

Three large national data collection companies (referred to as A, B, and C) were contacted for bids. They made preliminary investigations on the feasibility of collecting responses from individuals who had engaged in an agile project that had 20 or more engaged members. All three companies were upfront about the difficulty in achieving a target sample size of 125. Company A came up with an incidence response rate of 15–25%, and companies B and C came up with an incidence response rate of about 25%. The incidence

response rate is similar to the response rate but is the terminology used by data collection companies to set charges per respondent. After conducting phone interviews with the sales manager of the three companies, the researcher selected Toluna, company B, because of the confidence in the capability of the company to deliver the sample size that met the requirements and filters. The company subscribes to the data collection code of standards and ethics of organizations such as ESOMAR and Insights Association. The company delivered the data in three weeks. The researcher used personal funds for data collection payment.

The Qualtrics software was used to compose the survey (see Appendix A). After the data was collected, the incidence response rate came out as 32.2%. The valid response count was 123. Out of the 123 responses, 122 came from distinct IP addresses, thereby affirming that the data came from a reasonably random sample from the population. The sample size of 123 respondents is an adequate number based on power analysis (Hair et al., 2016) that for five independent constructs, the recommended sample size for detection of 10% R-square at a significance level of $p = 0.05$ is 122. If two single-item variables, experience, and organization size, are considered, then the sample size requirement increases to 137. Past studies involving perceived usefulness have reported a much higher R-square and, thus, the sample size was deemed to be adequate.

Table 1 provides information on project size, organization size, and agile experience. The median project size, in terms of engaged members, was 30. Note that about 85% of the project size count is in the range 20–50. The organization size shows a typical distribution, which is considerably skewed at the higher end; thus, its logarithm was used for analysis. Agile experience seems robust, with 53.6% of respondents having more than six years.

Table 2 shows the respondent profile by method, role, and industry. The respondent profile indicates a varied set of roles, but there is a somewhat larger representation of the project and senior managers as well as a fair representation of developers/programmers, software architects, systems analysts, Product Owners, and Scrum Masters. An assortment of industries is represented. The related software, telecommunication, and analytics had been combined into one category, which emerges as the dominant industry.

Although the most commonly used method is SAFE, there is a fair representation of other methods. The next employed method is company-specific agile, which suggests that there is possible customization of the agile methods for large projects. Agile-structured hybrid methods are also commonly used, suggesting that companies may have been able to balance structured methods like Rational Unified Process (RUP) with an agile approach like Scrum. The use of Scrum in LeSS, as well as Scrum of Scrums along with standalone Scrum, indicates the influence of Scrum for LSAM.

About 44% of the methods (corresponding to company-specific, agile hybrid, and agile-structured hybrid) do not fall under named categories. The finding suggests that companies are likely selecting suitable practices from different methods. On this issue, Jacobson and Stimson (2018) remark that even with the rise of agile, people still flip-flop from one branded approach to another,

Table 2
Respondent profile by method, role, and industry.

Method		Role		Industry	
Method name	Count by percent	Role name	Count by percent	Industry name	Count by percent
SAFe	28.5	Project Manager	26.8	Software/Telecom/Analytics	35.0
Company-Specific Agile	17.9	Senior Manager	22.8	Professional Services	13.8
Agile Hybrid	13.0	Developer/ Programmer	10.6	Manufacturing	12.2
Agile-Structured Hybrid	13.0	Software Architect	9.8	Retail	9.8
Large Scale Scrum (LeSS)	8.9	Systems Analyst	9.8	Financial Services	8.9
Scrum	8.1	Product Owner	6.5	Utilities	5.7
Kanban	5.7	Scrum Master	4.9	Other	4.1
Scrum of Scrums	4.1	Business Analyst	4.1	Healthcare	3.3
Some other Agile	0.8	Other	1.6	Education/Research	2.4
		Customer	1.6	Transportation	2.4
		Consultant/ Trainer	1.6	Media/Entertainment	0.8
				Marketing	0.8
				Government	0.8

throwing away the good with the bad and behaving more like religious cultists than like scientists. They recommend that the better option for organizations and teams is to be free to select and combine practices. Table 2 suggests that companies are indeed combining practices from different methodologies to meet the specific challenges of the applications; however, the customization does not imply that there is no method. Ultimately, the purpose of a method is to improve the software development process regarding the increased productivity of information technology (IT) personnel and the higher quality of the final IT solutions (Chan and Thong, 2009).

6. Initial and final research models

6.1. Initial research model

The significant factors identified in the previous section provide a basis for creating an initial structural equation model for empirical evaluation. Fig. 1 shows the initial relationships among the latent constructs (the inner model) and the indicators of each construct (the outer model). The initial research model includes the latent constructs but excludes the two single-item variables, experience and organization size, (which, in any case, had low correlations ranging from -0.16 to 0.22). Because all the indicators had adequate factor loadings (from 0.71 to 0.87), and satisfactory reliability and convergent validity as shown later in this section, the indicators of the latent variables are omitted under hypotheses and results, which illustrate the inner model only. Appendix A provides the correspondence between each indicator and its corresponding description.

As Fig. 1 displays, perceived usefulness was considered as the critical determinant of methodology acceptance. Compatibility, subjective norm, and external support were considered as antecedents to both perceived usefulness and LSAM acceptance while perceived usefulness and mandatoriness were regarded as the antecedents to LSAM acceptance. The centrality of perceived usefulness is ascribed to its dominant role in most of the literature described in the previous section. Since the (Davis, 1989) seminal paper on Technology Acceptance Model (TAM), numerous acceptance studies have shown that perceived usefulness is the critical determinant of behavioral intention to accept innovation. In the five research models considered by Riemenschneider et al. (2002) study, the coefficient explaining the effect of perceived usefulness on the intention to use accounted from 0.48 – 0.72 . The next highest coefficients were subjective norm/social factors, mandatoriness/voluntariness, and compatibility with the range 0.21 – 0.28 . None of the many other variables were significant. The magnitude of the impact of perceived usefulness is invariably high in most in-

formation systems (IS) studies. The (Vijayasarathy and Turk, 2012) study is somewhat of an outlier because it shows a moderating but not a direct effect of perceived usefulness.

Karahanna and Straub (1999) investigated the psychological antecedents, and empirically demonstrated that perceived usefulness has antecedents other than perceived ease-of-use (PEOU). They found that social influence, which is equivalent to social or subjective norms, is a significant precursor. They also hypothesized that user training and support is an antecedent, but perhaps because the domain was email use, which needs a small level of training for gaining proficiency, the relationship was found to be not significant in their study. Karahanna and Straub (1999) admitted that it is conceivable that training and support affect perceptions and usage during initial usage of the technology, but their importance declines with continued use of the technology. In the case of LSAM, which is a complex domain, training and support can be an important antecedent of perceived usefulness, given that it takes a considerable time for gaining proficiency.

The relationship between compatibility and perceived usefulness is of particular interest in this study. A meta-analysis of 75 studies revealed that relative advantage and compatibility are the two of the three most important factors in predicting adoption and implementation of innovation (Tornatzky and Klein, 1982). The degree of relative advantage is equivalent to perceived usefulness (Riemenschneider et al., 2002). The related literature has shown the relationship between compatibility and perceived usefulness in different ways. Rogers (1983) shows the two as separate constructs that affect acceptance, whereas Moore and Benbasat (1991) infer that the two are not distinct constructs and can be merged into perceived usefulness. There is a fair number of studies that consider compatibility as an antecedent of perceived usefulness (Lee et al., 2011; Templeton and Byrd, 2003; Wu and Wang, 2005). Some studies indicate that the two are highly correlated (Taylor and Todd, 1995; Templeton and Byrd, 2003), and, thus, one may conjecture that the two constructs can be aspects of a higher-order construct. In this study, the researcher chose to show compatibility as an antecedent of perceived usefulness in the initial model.

6.2. Reliability

The PLS algorithm was employed for conducting the initial data analysis. The reliability and convergent validity were acceptable (see Table 3). Reliability refers to the general agreement between multiple items and can be measured using Cronbach's alpha or composite reliability. Cronbach's alpha provides an estimate of the reliability based on the intercorrelations of the observed indicator variables (Hair et al., 2016). The desirable value for Cronbach's al-

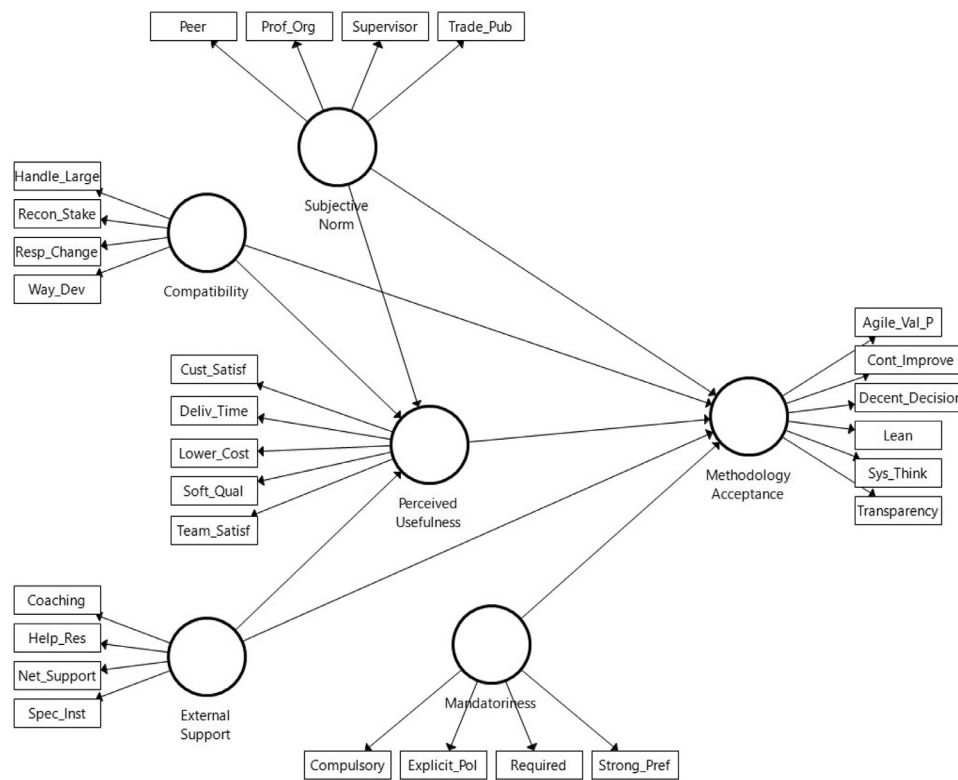


Fig. 1. The initial research model.

Table 3
Reliability and convergent validity.

	Cronbach's alpha	Composite reliability	Convergent validity (AVE > 0.5)
Compatibility	0.873	0.913	0.724
External support	0.861	0.906	0.706
Mandatoriness	0.801	0.862	0.611
Methodology acceptance	0.893	0.918	0.651
Perceived usefulness	0.877	0.910	0.670
Subjective norm	0.771	0.853	0.594

pha is over 0.70. The constructs in the study showed Cronbach's alpha ranging from 0.77 to 0.89. Composite reliability is considered an alternative and better measure of reliability, with the recommended range as 0.8–0.9 (Hair et al., 2016). The values ranged from 0.85 to 0.92 and were significant at p -value = 0.000. Thus, the reliability of the constructs was established.

6.3. Convergent validity

Convergent validity is the extent to which a measure correlates positively with other measures of the same construct (Hair et al., 2016). In assessing convergent validity, indicators of a construct are treated as alternative approaches for measuring the same construct by calculating loadings. Indicator reliability was estimated by calculating the outer loadings, which ranged from 0.708 to 0.872. An outer loading of less than 0.7 is usually not acceptable (Hair et al., 2016). All outer loadings were significant at p -value = 0.000 (which implies $p < 0.001$). Thus, the indicator reliability of the scales was established. A standard measure to verify the convergent validity at the construct level is the average variance extracted (AVE) (Hair et al., 2016), which should be above 0.5. As shown in Table 3, the AVE for the constructs varied from 0.594 to 0.724. Thus, the convergent validity of the constructs was established.

6.4. Discriminant validity

The logic of discriminant validity is based on the idea that a construct shares more variance with its associated indicators than with any other construct (Hair et al., 2016). Although discriminant validity was technically established, there was one issue: the correlation between perceived usefulness and compatibility was 0.803 while the square root of average variance extracted (AVE) of perceived usefulness was 0.818, and that of compatibility was 0.851. In other words, the discriminant validity was established only marginally for this one case, and although the correlation was not extreme, it did raise some concern about the accurate estimation of structural-order relationships. A revised research model corrected the shortcoming.

6.5. Revised research model

According to Hair et al. (2016), there may be sound reasons to include a hierarchical-component model (HCM) in a PLS path model if the first-order constructs are highly correlated, thereby resulting in inaccurate estimations of structural-order relationships. The measures of the perceived usefulness had some similarities with those of compatibility. According to (Riemenschneider et al., 2002), perceived usefulness corresponds with job fit, that is, the goals of the project while compatibility

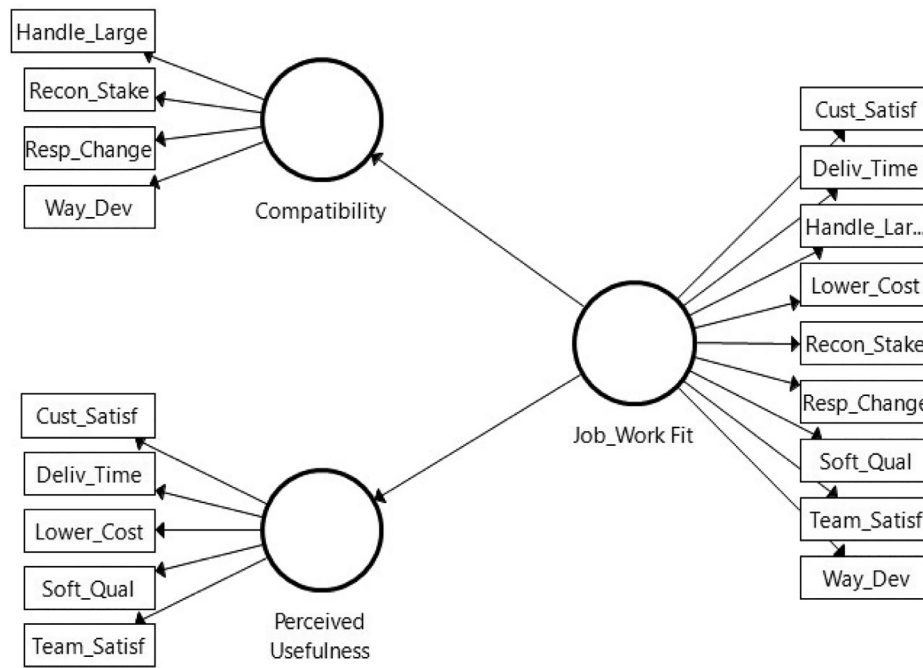


Fig. 2. Research model showing job-work fit as higher-order.

Table 4
Discriminant validity.

	External support	Mandatoriness	Job-work fit	Methodology acceptance	Subjective norm
External Support	0.840				
Mandatoriness	0.562	0.782			
Job-work fit	0.766	0.500	0.949		
Methodology Acceptance	0.509	0.467	0.588	0.807	
Subjective Norm	0.512	0.577	0.679	0.502	0.770

corresponds with work fit, that is, how is the method compatible with individual's manner of working. The constructs usefulness/job fit, and compatibility/ work fit were, thus, defined as part of "job-work fit," which was considered as a second-order construct. Job-work fit can be defined as the adopter's perception that the practices of the software development method facilitate the economic and social expectations of the job, and match the aspects of the work requirements.

Fig. 2 demonstrates how a second-order construct is shown in PLS. Normally, a construct has its own indicators. A higher-order construct is defined as containing all the indicators of its lower-order constructs. Thus the higher-order construct job-work fit contains the indicators of compatibility and perceived-usefulness. In Fig. 2, the direction of the arrows from job-work fit to compatibility and perceived usefulness suggests that the lower-order measures are reflective, not formative.

The revised model was run in two steps. In the first step, the initial model was modified by using the job-work fit as a higher-order mediating variable consisting of compatibility and perceived usefulness. From the PLS results, the latent values of compatibility and perceived usefulness were obtained. In the second step, the same model was rerun (see Fig. 3) using the latent values of compatibility and perceived usefulness. Note that Fig. 3 shows the latent value indicators of job-work fit.

As expected, the creation of a higher-order construct resolved the discriminant validity issue (see Table 4). For readability, Fig. 3 does not show the indicators of constructs other than job-work fit; the sign [+] implies that these can be revealed by expanding the construct. Fig. 1 already provides the indicator names,

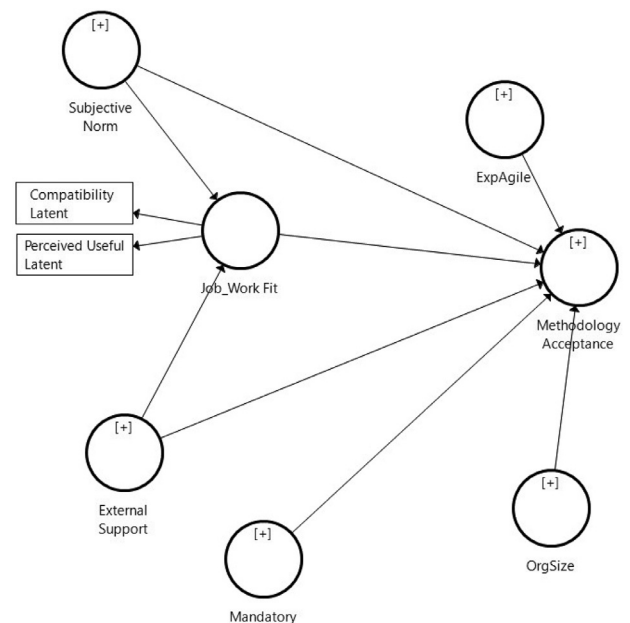


Fig. 3. Research model with the job-work fit as a higher-order construct.

which can be mapped to the questionnaire items included in Appendix A. Note that agile experience and organization size are single-item demographic variables.

Table 5
Structural model paths and the corresponding hypotheses.

Path	Hypotheses
Subjective Norm -> Job-work fit	H1: There is a positive association between subjective norm and job-work fit
External Support -> Job-work fit	H2: There is a positive association between external support and job-work fit
Job-work fit -> Methodology Acceptance	H3: There is a positive association between job-work fit and methodology acceptance
Subjective Norm -> Methodology Acceptance	H4: There is a positive association between subjective norm and methodology acceptance
External Support -> Methodology Acceptance	H5: There is a positive association between external support and methodology acceptance
Mandatoriness -> Methodology Acceptance	H6: There is a positive association between mandatoriness and methodology acceptance
Agile Experience -> Methodology Acceptance	H7: There is a positive association between agile experience and methodology acceptance
Organization Size -> Methodology Acceptance	H8: There is a positive association between organization size and methodology acceptance

Table 6
Path coefficients and significance of hypotheses.

Path	Path coefficient	p-value	Hypothesis
Subjective Norm -> Job-work fit	0.390	0.000	Supported
External Support -> Job-work fit	0.565	0.000	Supported
Job-work fit -> Methodology Acceptance	0.379	0.005	Supported
Subjective Norm -> Methodology Acceptance	0.108	0.573	Not Supported
External Support -> Methodology Acceptance	0.044	0.793	Not Supported
Mandatoriness -> Methodology Acceptance	0.171	0.248	Not Supported
Agile Experience -> Methodology Acceptance	0.105	0.053	Supported
Organization Size -> Methodology Acceptance	-0.030	0.760	Not Supported

6.6. Hypotheses

The paths shown in Fig. 3 denote the hypotheses listed in Table 5. Each path represents a potential positive or negative association. Sections 6.1 and 6.5 provide the reasoning for the paths among the primary constructs (hypotheses H1-H6); consequently, the justification is not repeated. In the revised model, two single-item variables are added for testing hypotheses H7 and H8. Given that LSAM relies considerably on agile methods like Scrum, the association between agile experience and methodology acceptance (H7) is hypothesized as positive. Bigger organizations are likely to have more extensive projects and, thus, the association (H8) is hypothesized as positive.

Note that the hypotheses shown in Table 5 show direct effects. As an aside, one may also want to test indirect effects for two cases: subjective norm and external support. For example, the indirect effect of the subjective norm to methodology acceptance is the product of the direct effects H1 (subjective norm -> job-work fit) and H3 (job-work fit -> methodology acceptance). Similarly, the indirect effect of external support to methodology acceptance is the product of the direct effects of H2 and H3. Because the direct paths are already included in the hypotheses, the indirect effects are provided as additional, derived results.

7. Results

Fig. 4 shows the results from the analysis by using the Smart PLS version 3 (Ringle et al., 2015). The R-square for job-work fit was 0.70, and for methodology acceptance was 0.40. The thickness of the lines in the figure is proportional to the coefficient and reveals the significant paths. External support has a strong effect, while subjective norm has a moderate effect on job-work fit, which then has a substantial impact on methodology acceptance. Thus, job-work fit mediates the effects of subjective norm and external support and emerges as the central determinant of methodology acceptance. Subjective norm, external support, and mandatoriness do not have significant direct effects on methodology acceptance.

Table 6 lists the coefficients and significance of the hypotheses. The p-value 0.000 means that it is smaller than 0.001, which suggests strong significance. For the relationships among the key constructs, hypotheses H1, H2, and H3 are supported, but H4, H5, and H6 are not supported. Furthermore, external support has a strong

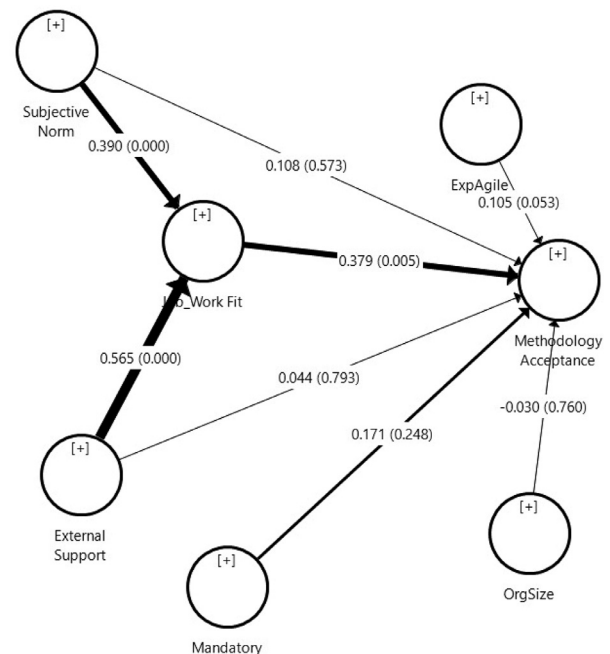


Fig. 4. SmartPLS results from the revised research model.

indirect effect ($p = 0.004$), and the subjective norm has a marginal indirect effect ($p = 0.058$).

Agile experience and organization size were each measured using a single indicator. The respondent's experience with the agile methodology was hypothesized to have a positive effect on methodology acceptance (hypothesis H7). The reasoning is that the familiarity with the agile methodology will result in better acceptance of LSAM. The effect was considered as significant at a p-value of 0.053, although technically, the p-value is just above the cutoff level of 0.05, and the coefficient is weak. Thus, the variable agile experience needs further research. The results indicated that organization size had practically no effect on the LSAM acceptance (hypothesis H8). A post-hoc test was run by partitioning the data by role into two sets – managers and non-managers. The results indicate that there is practically no difference between the strengths

of the relationships of mandatoriness with acceptance. For managers, the coefficient is 0.14 while for non-managers, the coefficient is 0.17. Neither coefficient is significant at $p = 0.05$.

8. Discussion

The research delved into the essential research question: which factors determine the acceptance and use of large-scale agile methodology (LSAM)? The study considered the following constructs as affecting LSAM acceptance: perceived usefulness, compatibility, subjective norm, mandatoriness, and external support. These constructs have been predominantly found as significant in empirical studies. In addition to the primary constructs, two other variables, respondent experience with agile methodology and the size of the organization, were considered. The median size of the project in terms of engaged members was 30, which corresponds to about four teams (Dingsøyr et al., 2014).

8.1. Implications for theory

In this study, perceived usefulness and compatibility were strongly correlated. Perceived usefulness is equivalent to job fit, while compatibility is equivalent to work fit (Riemenschneider et al., 2002; Thompson et al., 1991). The study proposed job-work fit as a higher-order construct that considered job fit (perceived usefulness) and work fit (compatibility) as components. This formulation is a unique contribution to the research. The results indicate that job-work fit is a strong predictor of LSAM methodology acceptance and mediates the effects of subjective norm and external support. Respondents will likely evaluate LSAM acceptance if they perceive that the method will provide a distinct improvement in software quality, delivery time, lowering cost, customer satisfaction, and team satisfaction and that the technique will align with their work practices in handling large projects, responding to changes in requirements, and reconciling coordination issues. Other factors such as subjective norm and external support enhance job-work fit, which in turn increases LSAM acceptance.

Methodology acceptance was measured in terms of adherence to the vital characteristics of LSAM. The approach, although somewhat novel in the methodology acceptance domain, has precedence and is similar to that followed by Vijayasarathy and Turk (2012), although their study mixed methods and practices when measuring agile use. The approach is consistent with the recommendations of (Tornatzky and Klein, 1982) that acceptance is measured by considering the critical elements of a diffusion. To measure LSAM acceptance, it is essential to distill its practices to the fundamental values and principles. Research and books such as Bick et al. (2018), Conboy (2009), Knaster and Leffingwell (2017), Larman and Vodde (2016) were used as the primary guide in selecting the following LSAM characteristics: agile values and principles, systems thinking, lean product development, transparency, continual improvement, and decentralized decision making. The interpretation of LSAM is much broader than that of agile development. The approach based on adherence to essential values minimizes the problem that a respondent claims that she followed LSAM but may not have implemented principles such as lean development and systems thinking; instead, she may have just shadowed agile practices such as pair programming and stand-up meetings. A literal adoption of a named method is not equivalent to the acceptance of its underlying values (Conboy, 2009). Even the Agile Manifesto (Fowler and Highsmith, 2001) starts with four fundamental defining values, while particular methods implement the Agile Manifesto in specific but similar ways.

The results from this study overlap those of other methodology acceptance studies such as Vijayasarathy and Turk (2012)

and Riemenschneider et al. (2002); however, there are some critical differences. For example, the (Vijayasarathy and Turk, 2012) study emphasized subjective norm and training while the (Riemenschneider et al., 2002) study accentuated perceived usefulness with emphasis on subjective norm, compatibility, and mandatoriness. In this study, except for mandatoriness, the remaining constructs had a role in methodology acceptance. There were no mediating constructs in the past software development acceptance studies. The job-work fit is not only a direct predictor of acceptance, but it also mediates the relationship from subjective norm and external support to methodology acceptance. In the LSAM domain, engaged participants not only value the benefits of their job, but they simultaneously desire harmony with their work practices. Some variables indirectly influence methodology acceptance; for example, the study found that external support has a significant indirect effect even as it had no direct impact on LSAM acceptance.

The insignificance of mandatoriness in this study is evident once we examine the LSAM aspects such as agile values and principles, transparency, and decentralized decision-making, which conflict with mandatoriness, which evokes a top-down dictate as may have been acceptable in structured ways of working twenty or more years ago. In today's software development environment, a developer is likely to select a work environment that is conducive to self-organization (Hoda et al., 2013) and personal values, free-will, and agency. In a study on organizations undergoing an agile and lean transformation, Power (2016) found that people perceive there is too much autonomy rather than too little; consequently, there is not much room for mandatoriness. In the study, the respondents had completed a project using a large-agile methodology. They were, therefore, aligned with precepts that value individuals over processes even if the organization might have encouraged a brand of LSAM through a directive.

Subjective norm was the key finding of the (Vijayasarathy and Turk, 2012) study, which focused on agile adoption and use. One plausible explanation is that their research was conducted when agile development was still reaching maturity. As of 2018, LSAM has not reached maturity, although agile development has become commonplace. Because the popular LSAM such as SAFe, LeSS, and Scrum of Scrums are usually based on Scrum (Schwaber, 2004), the general awareness of LSAM may have contributed to a less than strong effect of subjective norm in the current study.

The results of this study are consistent with (Frambach and Schillewaert, 2002), which articulated that the acceptance of innovation is dependent on personal characteristics such as tenure and product experience. Software development experience had a negative effect on methodology acceptance in the (Vijayasarathy and Turk, 2012) study, while experience with agile development has a positive impact in this study. The difference between the two studies is in the kind of experience. It seems that the software development experience in the case of (Vijayasarathy and Turk, 2012) research could have been on structured methodologies, which have a different underlying philosophy as compared with agile methods (Nerur et al., 2005). In the current study, the experience with agile development, which is better harmonized with LSAM development, acted as a positive influence on the respondents. As shown in Table 2, the popular methods used for LSAM build on agile concepts.

The respondent profile indicates that there is a variety of LSAM methods. Although SAFe was the most commonly used method, the next five employed methods were: company-specific agile, agile hybrid, agile-structured hybrid, LeSS, and Scrum. Together these methods represented about 90% of the cases. Thus, methods such as Scrum (generally associated with agile) and agile-structured hybrid (such as an agile version of RUP) could qualify as LSAM methods. The field may witness a better operationalization of LSAM in the next few years.

8.2. Implications for the practitioners

The practitioner interested in the acceptance of an LSAM methodology may want to focus on two key elements: perceived usefulness (i.e., job fit) and compatibility (i.e., work fit). If these elements are missing, then other factors will not be impactful; thus, the combined job-work fit is the gateway for other factors. In this domain, respondents view job fit as allied with work fit. For example, a senior manager may evaluate and emphasize how the LSAM method can improve product quality, lower the cost, manage the schedule, and increase customer and employee experiences. She may also want to evaluate whether the values and work practices underlying the method rhyme with the work experiences of the employees, such as developers and product managers. After such evaluations, she can then design mechanisms for spreading the usefulness and compatibility themes through social channels such as wikis or meetups. Simultaneously, she needs to provide external support through training and availability of champions and consultants. Practitioners who have agile development experience are more open to acceptance of LSAM and should be encouraged to champion the diffusion of the merits of job-work fit.

The underlying values of LSAM are inconsistent with that of a mandatory, hierarchical culture; consequently, organizational mandates should be minimized. In this respect, [Iivari and Huisman \(2007\)](#) recommend that an organization should pay special attention to means of avoiding the hierarchical flavor of SDMs when introducing them by engineering an SDM that is less bureaucratic rather than as a complicated conglomerate of numerous techniques with extensive documentation. The current study indicates that both managers and non-managers are aligned with the more straightforward, less bureaucratic LSAM approach. Overall, practitioners do not care much about the exact name of an LSAM, but are more concerned about the value delivered ([Conboy and Carroll, 2019](#)); in this regard, the study focuses on the vital LSAM elements rather than on specific names and practices.

9. Limitations of the study

The main limitation of the study is that the combining of perceived usefulness and compatibility in this study is based on a statistical instead of a theoretical basis. This approach is acceptable if we deem that the interpretation of a construct depends on the context, and two constructs may become closer in specific domains (e.g., LSAM) resulting in the use of a higher-order approach. The interpretation of perceived usefulness as job fit and of compatibility as work fit with slightly overlapping meaning led to the treatment of the two constructs as items of a higher order construct, which was named as job-work fit, for the methodology acceptance domain. This approach is statistically sound but necessitates replication of the study after a closer examination of construct indicators. Furthermore, a conceptual or theoretical grounding of the higher-order construct, job-work fit, is desirable.

The use of a professional data collection company can be considered both as a strength and a limitation. The strength is obtaining a reasonable sample from separate contexts: large projects defined as having 20 or more engaged members (e.g., developers, product managers) are relatively rare. It is not difficult to find one or a few organizations that have large projects, but if the analysis is at the individual instead of the project level, then the methodology is weakened because of systemic bias stemming from common characteristics of the sample. Individuals that belong to the same project, and the same company will tend to fill the survey in a similar fashion. For example, if a company has an organizational mandate on the use of an LSAM methodology, then we may find mandatoriness as a significant factor, although the effect may be spurious. A researcher usually lacks the resources to conduct

a nationwide search so that the responses are based on different projects and, thus, may have to rely on a professional data collection agency. However, the data gathering has a limitation: the reliance on a third party for the collection of quality data, despite the quality checks, results in less control of the process. Overall, the use of a third party was likely justified, but future research may consider alternative data collection approaches.

As demonstrated in the methodology section, the sample size of 123 is adequate for the overall research; however, it is not sufficient if the post-hoc analysis is desired based on splitting the sample on different values of a demographic variable such as project size, role, or industry. For example, an examination of roles suggests that the sample is more heavily weighted toward managers. Bigger sample size would have provided adequate statistical power for post-hoc analysis of managers versus non-managers. The verification of 122 IP addresses does not imply that the sample included the same number of distinct projects, but it does suggest that there was sufficient variety as is supported by the demographics shown in [Tables 1 and 2](#). Ultimately, the study design does not rely so much on a high number of participants, but from a reasonable count of separate contexts.

The construct measures are acceptable on several criteria, such as factor loadings, reliability, convergent validity, and discriminant validity. However, construct measures were adapted and improvised based on the domain characteristics, and the passing of validity tests does not imply the constructs are complete and final.

10. Conclusion

This LSAM acceptance study reconciles the findings between the research on acceptance of agile and structured development methodologies. A pivotal agile development acceptance study ([Vijayasathiy and Turk, 2012](#)) emphasized the roles of subjective norm and training while a crucial structured development acceptance study ([Riemenschneider et al., 2002](#)) underscored the importance primarily of perceived usefulness and secondarily of compatibility, subjective norm, and organizational mandate. An LSAM is based mainly on agile development but has some elements of structured development. The results indicate that perceived usefulness and compatibility are aspects of a higher order construct, job-work fit, which has a direct effect on LSAM acceptance. Subjective norm and external support do not have direct impacts on LSAM acceptance but have direct effects on job-work fit, which is a central construct that mediates the relationships. Future research may have to take a confirmatory approach. A complementary next research theme, which can be accomplished by using qualitative methods, would be to examine the earlier phases of adoption of LSAM. Case studies of successful or unsuccessful attempts to institutionalize LSAM may reveal the critical factors essential at the different stages of acceptance.

Declaration of Competing Interest

Title of Manuscript: Job-Work Fit as a Determinant of the Acceptance of Large-Scale Agile Methodology.

CRediT authorship contribution statement

Dinesh Batra: Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing - original draft, Writing - review & editing.

Acknowledgments

The author thanks the associate editor and the anonymous reviewers for their very useful feedback and suggestions.

Appendix A

Large-scale agile methodology acceptance questionnaire

Note that the respondent selected a methodology such as SAFe or Scrum from a list presented earlier (see Table 2). The selected methodology was coded in the questionnaire and is included here as <method name from a previous choice>.

Large-scale agile methodology acceptance

The construct was measured by using the following Likert scale
None at all (1) A little (2) To some extent (3) A moderate amount (4) Significantly (5) Highly (6) Very Highly (7)

To what extent the following values, principles, practices, or aspects are actually practiced in your use of the adopted LSAM <method name from a previous choice>:

- (Agile_Val_P). Agile values and principles
- (Sys_Think). Systems thinking
- (Lean). Lean product development
- (Transparency). Transparency
- (Cont_Improve). Continual improvement
- (Decent_Decision). Decentralized decision making

The constructs subjective norm, perceived usefulness, mandatoriness, external support, and compatibility were measured using the following Likert scale:

Strongly disagree (1) Disagree (2) Somewhat disagree (3) Neither agree nor disagree (4) Somewhat agree (5) Agree (6) Strongly agree (7)

Subjective norm

To what extent have the following influenced your acceptance of the large-scale agile methodology (LSAM) <method name from a previous choice>?

- (Peer). Peer groups
- (Supervisor). Supervisors
- (Trade_Pub). Trade Publications
- (Prof_Org). Professional Organizations

Perceived usefulness

As compared with alternative methodologies, I perceive that the adopted LSAM <method name from a previous choice> provides a distinct improvement in

- (Soft_Qual). Software quality
- (Deliv_Time). Delivery Time
- (Lower_Cust). Lowering costs
- (Cust_Satisf). Customer satisfaction
- (Team_Satisf). Team satisfaction

Mandatoriness

Regarding the adopted large-scale agile methodology (LSAM) <method name from a previous choice> mentioned by you:

- (Compulsory). Using the LSAM is compulsory in my job (Required). Although I might prefer another methodology, I am required to use the LSAM
- (Explicit_Pol). The company has an explicit policy for using the LSAM
- (Strong_Pref). The company has a strong preference for standardizing development using the LSAM

External support

For helping me to better understand and use the adopted LSAM <method name from a previous choice>, the following resources are widely available:

- (Spec_Inst). Specialized instruction
- (Coaching). Coaching resource
- (Net_Support). Network of support
- (Help_Res). Help and resources

Compatibility

I feel that using the adopted LSAM <method name from a previous choice> is more compatible than other methodologies for

- (Way_Dev). The way I develop systems
- (Handle_Large). Handling large projects
- (Recon_Stake). Reconciling different views of stakeholders
- (Resp_Change). Responding to changes in requirements

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at [doi:10.1016/j.jss.2020.110577](https://doi.org/10.1016/j.jss.2020.110577).

References

- Ajzen, I., 1991. The theory of planned behavior. *Organ. Behav. Hum. Decis. Process.* 50 (2), 179–211.
- Alqudah, M., Razali, R., 2016. A review of scaling agile methods in large software development. *Int. J. Adv. Sci. Eng. Inf. Technol.* 6 (6), 828–837.
- Barki, H., Rivard, S., Talbot, J., 1993. Toward an assessment of software development risk. *J. Manag. Inf. Syst.* 10 (2), 203–225.
- Beck, K., 2000. *Extreme Programming Explained*. Addison-Wesley, Boston, MA.
- Bick, S., Spohrer, K., Hoda, R., Scheerer, A., Heinzl, A., 2018. Coordination challenges in large-scale software development: a case study of planning misalignment in hybrid settings. *IEEE Trans. Softw. Eng.* 44 ((10)), 932–950.
- Björnson, F.O., Wijnmaalen, J., Stettina, C.J., Dingsøyr, T., 2018. Inter-team coordination in large-scale agile development: a case study of three enabling mechanisms. In: *Proceedings of the International Conference on Agile Software Development*.
- Boehm, B.W., Turner, R., 2004. *Balancing Agility and Discipline: A Guide For the Perplexed*. Addison-Wesley, Boston, MA.
- Cao, L., Mohan, K., Xu, P., Ramesh, B., 2009. A framework for adapting agile development methodologies. *Eur. J. Inf. Syst.* 18 (4), 332–343.
- Chan, F.K., Thong, J.Y., 2009. Acceptance of agile methodologies: a critical review and conceptual framework. *Decis. Support Syst.* 46 (4), 803–814.
- Conboy, K., 2009. Agility from first principles: reconstructing the concept of agility in information systems development. *Inf. Syst. Res.* 20 (3), 329–354.
- Conboy, K., Carroll, N., 2019. Implementing large-scale agile frameworks: challenges and recommendations. *IEEE Softw.* 44–50. doi:10.1109/MS.2018.2884865.
- Davis, F.D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.* 13 (3), 319–340.
- Davis, F.D., Bagozzi, R.P., Warshaw, P.R., 1989. User acceptance of computer technology: a comparison of two theoretical models. *Manage. Sci.* 35 (8), 982–1003.
- Diamantopoulos, A., Siguaw, J.A., 2000. *Introducing LISREL: A Guide for the Uninitiated*. SAGE, London, UK.
- Dikert, K., Paasivaara, M., Lassenius, C., 2016. Challenges and success factors for large-scale agile transformations: a systematic literature review. *J. Syst. Softw.* 119, 87–108. doi:10.1016/j.jss.2016.06.013.
- Dingsøyr, T., Fægri, T.E., Itkonen, J., 2014. What is large in large-scale? A taxonomy of scale for agile software development. In: Andreas Jedlitschka, P.K., Kuhrmann, Marco, Männistö, Tomi, Münch, Jürgen, Raatikainen, Mikko (Eds.). *Product-Focused Software Process Improvement*, 8892. Springer International Publishing, pp. 273–276.
- Dingsøyr, T., Faleisi, D., Power, K., 2019. Agile development at scale: the next frontier. *IEEE Softw.* 30–38. doi:10.1109/MS.2018.2884884.
- Dingsøyr, T., Moe, N.B., Fægri, T.E., Seim, E.A., 2018. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empir. Softw. Eng.* 23 (1), 490–520.
- Downs, G.W., Mohr, L.B., 1976. Conceptual issues in the study of innovation. *Adm. Sci. Q.* 21 (4), 700–714.
- Dyba, T., Dingsøyr, T., 2008. Empirical studies of agile software development: a systematic review. *Inf. Softw. Technol.* 50 (9–10), 833–859.
- Ebert, C., Paasivaara, M., 2017. Scaling agile. *IEEE Softw.* 34 (6), 98–103.
- El Emam, K., Koru, A.G., 2008. A replicated survey of IT software project failures. *IEEE Softw.* 25 (5), 84–90.
- Fichman, R.G., Kemerer, C.F., 1997. The assimilation of software process innovations: an organizational learning perspective. *Manage. Sci.* 43 (10), 1345–1363.

- Fichman, R.G., Kemerer, C.F., 1999. The illusory diffusion of innovation: an examination of assimilation gaps. *Inf. Syst. Res.* 10 (3), 255–275.
- Fowler, M., Highsmith, J.A., 2001. The agile manifesto. *Softw. Dev.* 9 (8), 28–35.
- Frambach, R.T., Schillewaert, N., 2002. Organizational innovation adoption: a multi-level framework of determinants and opportunities for future research. *J. Bus. Res.* 55 (2), 163–176.
- Gefen, D., Straub, D., Boudreau, M.-C., 2000. Structural equation modeling and regression: guidelines for research practice. *Commun. Assoc. Inf. Syst.* 4 (1), 7.
- Hair, J.F., Hult, G.T.M., Ringle, C., Sarstedt, M., 2016. *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*. Sage Publications, Thousand Oaks, CA.
- Hair, J.F., Ringle, C.M., Sarstedt, M., 2011. PLS-SEM: indeed a silver bullet. *J. Market. Theory Pract.* 19 (2), 139–152.
- Hair, J.F., Sarstedt, M., Ringle, C.M., Mena, J.A., 2012. An assessment of the use of partial least squares structural equation modeling in marketing research. *J. Acad. Market. Sci.* 40 (3), 414–433.
- Hannay, J.E., Benestad, H.C., 2010. Perceived productivity threats in large agile development projects. In: *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. Bolzano-Bozen, Italy.
- Hardgrave, B.C., Davis, F.D., Riemenschneider, C.K., 2003. Investigating determinants of software developers' intentions to follow methodologies. *J. Manag. Inf. Syst.* 20 (1), 123–151.
- Hardgrave, B.C., Johnson, R.A., 2003. Toward an information systems development acceptance model: the case of object-oriented systems development. *IEEE Trans. Eng. Manag.* 50 (3), 322–336.
- Hoda, R., Noble, J., Marshall, S., 2011. The impact of inadequate customer collaboration on self-organizing Agile teams. *Inf. Softw. Technol.* 53 (5), 521–534.
- Hoda, R., Noble, J., Marshall, S., 2012. Developing a grounded theory to explain the practices of self-organizing Agile teams. *Emp. Softw. Eng.* 17 (6), 609–639.
- Hoda, R., Noble, J., Marshall, S., 2013. Self-organizing roles on agile software development teams. *IEEE Trans. Softw. Eng.* 39 (3), 422–444.
- Iivari, J., Huisman, M., 2007. The relationship between organizational culture and the deployment of systems development methodologies. *MIS Q.* 35–58.
- Jacobson, I., Stimson, R., 2018. Tear down the method prisons!: set free the practices!. *Queue* 16 (5), 1–27.
- Johnson, R., 1999. Applying the technology acceptance model to a systems development methodology. In: *Proceedings of the AMCIS*.
- Kacmar, C.J., McManus, D.J., Duggan, E.W., Hale, J.E., Hale, D.P., 2009. Software development methodologies in organizations: field investigation of use, acceptance, and application. *Inf. Resour. Manag. J.* 22 (3), 16–39.
- Karahanna, E., Straub, D.W., 1999. The psychological origins of perceived usefulness and ease-of-use. *Inf. Manag.* 35 (4), 237–250.
- Khalifa, M., Verner, J.M., 2000. Drivers for software development method usage. *IEEE Trans. Eng. Manag.* 47 (3), 360–369.
- Knaster, R., Leffingwell, D., 2017. *SAFe 4.0 Distilled: applying the Scaled Agile Framework for Lean Software and Systems Engineering*. Addison-Wesley Professional, Boston, MA.
- Larman, C., Vodde, B., 2016. *Large-scale Scrum: More with LeSS*. Addison-Wesley Professional, Indianapolis.
- Lee, G., Xia, W., 2010. Toward agile: an integrated analysis of quantitative and qualitative field data. *Manag. Inf. Syst. Q.* 34 (1), 7.
- Lee, Y.-H., Hsieh, Y.-C., Hsu, C.-N., 2011. Adding innovation diffusion theory to the technology acceptance model: supporting employees' intentions to use e-learning systems. *J. Educ. Technol. Soc.* 14 (4).
- Leffingwell, D., 2007. *Scaling Software Agility: Best Practices For Large Enterprises*. Pearson Education, Upper Saddle River, NJ.
- Moe, N.B., Dingsøyr, T., Dyba, T., 2008. Understanding self-organizing teams in agile software development. In: *Proceedings of the 19th Australian Conference on Software Engineering ASWEC*.
- Moore, G.C., Benbasat, I., 1991. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Inf. Syst. Res.* 2 (3), 192–222.
- Nerur, S., Mahapatra, R., Mangalaraj, G., 2005. Challenges of migrating to agile methodologies. *Commun. ACM* 48 (5), 72–78.
- Østdahl, A. (2017). *Inter-team coordination in large-scale agile software development*. NTNU.
- Paasivaara, M., Lassenius, C., Heikkilä, V.T., 2012. Inter-team coordination in large-scale globally distributed scrum: do scrum-of-scrums really work? In: *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*.
- Pendharkar, P.C., Rodger, J.A., 2009. The relationship between software development team size and software development cost. *Commun. ACM* 52 (1), 141–144.
- Power, K., 2016. Sensemaking and complexity in large-scale lean-agile transformation: a case study from Cisco. In: *Proceedings the 49th Hawaii International Conference on System Sciences (HICSS)*.
- Richter, N.F., Cepeda, G., Roldán, J.L., Ringle, C.M., 2015. European management research using partial least squares structural equation modeling (PLS-SEM). *Eur. Manag. J.* 33 (1), 1–3.
- Riemenschneider, C.K., Hardgrave, B.C., Davis, F.D., 2002. Explaining software developer acceptance of methodologies: a comparison of five theoretical models. *IEEE Trans. Softw. Eng.* 28 (12), 1135–1145.
- Ringle, C.M., Wende, S., Becker, J.-M., 2015. *SmartPLS, 3*. SmartPLS GmbH, Boenningstedt.
- Roberts, T.L., Gibson, M.L., Fields, K.T., Rainer, R.K., 1998. Factors that impact implementing a system development methodology. *IEEE Trans. Softw. Eng.* 24 (8), 640–649.
- Roberts, T.L., Leigh, W., Purvis, R.L., Parzinger, M.J., 2001. Utilizing knowledge links in the implementation of system development methodologies. *Inf. Softw. Technol.* 43 (11), 635–640.
- Rogers, E.M., 1983. *Diffusion of Innovations*. Simon and Schuster, New York.
- Rolland, K.H., Fitzgerald, B., Dingsøyr, T., Stol, K.-J., 2016. Problematising agile in the large: alternative assumptions for large-scale agile development. In: *Proceedings of the Thirty Seventh International Conference on Information Systems*. Dublin.
- Scheerer, A., Hildenbrand, T., Kude, T., 2014. Coordination in large-scale agile software development: a multiteam systems perspective. In: *Proceedings of the 47th Hawaii International Conference on System Sciences*.
- Schwaber, K., 2004. *Agile Project Management With Scrum*. Microsoft Press.
- Strode, D.E., Huff, S.L., Hope, B., Link, S., 2012. Coordination in co-located agile software development projects. *J. Syst. Softw.* 85 (6), 1222–1238.
- Taylor, S., Todd, P.A., 1995. Understanding information technology usage: a test of competing models. *Inf. Syst. Res.* 6 (2), 144–176.
- Templeton, G.F., Byrd, T.A., 2003. Determinants of the relative advantage of a structured SDM during the adoption stage of implementation. *Inf. Technol. Manag.* 4 (4), 409–428.
- Theocharis, G., Kuhrmann, M., Münch, J., Diebold, P., 2015. Is Water-scrum-fall reality? On the use of agile and traditional development practices. In: *Proceedings of the International Conference on Product-Focused Software Process Improvement*.
- Thompson, R.L., Higgins, C.A., Howell, J.M., 1991. Personal computing: toward a conceptual model of utilization. *MIS Q.* 125–143.
- Tornatzky, L.G., Klein, K.J., 1982. Innovation characteristics and innovation adoption-implementation: a meta-analysis of findings. *IEEE Trans. Eng. Manag.* (1) 28–45.
- Van Waardenburg, G., Van Vliet, H., 2013. When agile meets the enterprise. *Inf. Softw. Technol.* 55 (12), 2154–2171.
- Venkatesh, V., Davis, F.D., 2000. A Theoretical extension of the technology acceptance model: four longitudinal field studies. *Manag. Sci.* 46 (2), 186–204.
- VersionOne. (2018). *The 12th annual state of agile survey*.
- Vijayasathy, L., Turk, D., 2012. Drivers of agile software development use: dialectic interplay between benefits and hindrances. *Inf. Softw. Technol.* 54 (2), 137–148. doi:10.1016/j.infsof.2011.08.003.
- Vinzi, V.E., Trinchera, L., Amato, S., 2010. PLS path modeling: from foundations to recent developments and open issues for model assessment and improvement. In: *Handbook of Partial Least Squares*. Springer, pp. 47–82.
- Wu, J.-H., Wang, S.-C., 2005. What drives mobile commerce?: an empirical evaluation of the revised technology acceptance model. *Inf. Manag.* 42 (5), 719–729.

Dinesh Batra is a professor in the Department of Information Systems and Business Analytics at Florida International University. Dr. Batra's publications have appeared in *Management Science*, *Journal of MIS*, *Journal of Systems and Software*, *Communications of the ACM*, *European Journal of Information Systems*, *Decision Support Systems*, *Information and Management*, *Journal of Database Management*, *Communications of the AIS*, *International Journal of Human-Computer Studies*, *Data Base for Advances in Information Systems*, *Requirements Engineering Journal*, *Information Systems Management*, and other journals. He is a co-author of the book *Object-Oriented Systems Analysis and Design* published by Pearson Prentice-Hall.