

**This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.**

Author(s): Kinnunen, Hanna; Luoma, Eetu

Title: Towards Measuring the Agility of Software Business

Year: 2018

Version:

Please cite the original version:

Kinnunen, H., & Luoma, E. (2018). Towards Measuring the Agility of Software Business. In Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS 2018) (pp. 5425-5434). University of Hawai'i at Manoa. Proceedings of the Annual Hawaii International Conference on System Sciences.
<https://doi.org/10.24251/hicss.2018.677>

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Towards Measuring the Agility of Software Business

Hanna Kinnunen
University of Jyväskylä
hanna.k.kinnunen@jyu.fi

Eetu Luoma
University of Jyväskylä
eetu.luoma@jyu.fi

Abstract

Agile development methods have been employed across the software industry. However, it is not always clear if the used methods actually help the software firms in being more agile and if agility has a positive influence in the software firm performance. Studying these questions may turn out impossible since good measurements for assessing the overall agility of software firms do not yet exist. A need is therefore detected to measure the differences in agility between firms and finding the means to evaluate the differences in agility in reliable manner. This article examines how to measure the agility of a software firm and reports initial steps in the process of developing measurement instruments. The measurement instrument is tested against data collected from Finnish software firms and purified for further analyses.

1. Introduction

In software development, agile methods are the means to adapt to changes in requirements rather than planning to avoid changes [1]. Agile methods have spread across the software industry from the beginning of the millennium. Their popularity can be detected from the requirements in job announcements, range of available tools as well as the interest of the research community. In fact, the amount of research articles concerning agile software development methods has been steadily increasing [1]. Agile methods have spread far and wide and fully plan-based methods seem to be a rarity [2]. It is therefore surprising that previous studies of agile methods have not dealt with the impact of applying agile methods to the performance of software firm and the software industry at large. Although agile methods have been studied extensively, no single empirical study exists which would consider whether agile methods are better for software business.

To understand the benefits of agile software development to software business and software firms, we must first be able to measure agility. Developing effective and valid measurement for evaluating

software firm's agility introduces an engaging question. While organizations strive to be agile by adopting methodologies such as Scrum and XP [3], projects, software firms and products are vastly different and the same practices are therefore applied very differently [4]. *How can we then measure the overall agility of a software firm?* Use of any single practice is easily observed, but research on software business would require a way to assess the firm's capability of being agile. When we are able to identify more or less agile software development firms, we can better understand basis for agility and the performance outcomes of agility.

In this article, we aim at understanding what agility is in software firms and how overall agility of a software firm can be measured. The questions are answered in the following sections. First, we discuss the meaning of agile software development and how agility has been studied in extant literature. We then develop a measuring instrument for evaluating the agility of software firms and present a purified measuring instrument based on findings of survey data. Lastly, we discuss the implications of the findings and give recommendations for future research.

2. Agility in software engineering and information systems development

Our review of literature was designed to discover how the agility concept is defined and understood in the information system and software engineering studies and to discover the possible ways to measure agility in the context of software firms. Acknowledging the vast amount of studies about the topic, we aimed at representative rather than comprehensive selection of studies and concept definitions. The starting point of our searches was therefore two literature reviews on agility published in 2008 [5] and 2012 [1]. Articles identified in these reviews were explored for definitions and measuring instruments of agility.

Agility is discussed here in the context of software engineering and development of information systems, where the term has been used since the late 1990s [6].

Agility and the agile philosophy have also been used in the context of manufacturing and administration for decades before they were adopted to software development [3]. The Manifesto for Agile Software Development (Agile Manifesto) was published in 2001 as a result of a workshop discussing new methods for software development. Many of the workshop participants were advocates for the XP-methods [7] but other methods were represented as well [8]. This is why it is easy to find similarities with the agile methods' values (XP, Scrum, Crystal) and the Agile Manifesto's principles [8, 9]. The Agile Manifesto presents four values for better software development and they are further divided into 12 principles. The four values are: "individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; responding to change over following a plan" [8].

There are varieties of other definitions, perhaps developed for the needs of different research domains. For instance an early definition of agility by Abrahamsson, Salo, Ronkainen and Warsta [11] stresses software engineering perspective. Their definition states that agile software development is based on cooperation, is adaptive to changes, but also produces functional software gradually or incrementally. Conboy [3] has created a definition for agility in information systems development (ISD), which is based on the assumption that agility is based on flexibility and leanness of operations. The definition is based on conceptual analysis of literature and is further developed into a taxonomy of ISD agility. It is worth quoting Conboy at length: Agility of an ISD method is "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." [3].

When we compare Conboy's definition of agility to the manifesto, it is obvious that the Agile Manifesto gives a much more detailed description on how agile software development should be done. On the other hand, Conboy's definition includes a temporal dimension and the business point of view is more prominent. Conboy uses a more scientific method in creating his definition and even refers to Agile Manifesto as "promotional material" which overly simplifies the matter [3]. Neither of these or any other definition have been widely accepted as the definition of agility, which leads to certain lack of theoretical

rigor in research. As long as agility cannot be defined, it is hard to understand how a software development method might contribute to a development team's ability to be agile [12]. Similarly, lack of functional measures makes it impossible to analyze how agility could contribute to performance of a software firm.

In order to develop useful means to evaluate the overall agility of a software firm, we must scrutinize the available definitions of the term. The following Table 1 analyzes articles where the definition of agility is discussed. The columns represent the components of the definition of agility used in the article. The first columns about "Agile Manifesto" and "Conboy's definition" denote that the succeeding articles have used the prior definition. The descriptions were searched from 20 articles. Agile manifesto was mentioned in 14 of them and the Conboy's definition in four out of eight that were published after the definition. The most common description of agility was reacting to change (in 17 of the 20 articles) and iterative and incremental releases (in 15 of the 20 articles). Speed and customer collaboration were mentioned in 12 articles and face-to-face and continuous communication was mentioned in ten articles. Flexibility and leanness were mentioned in eight articles.

When we examine the prevalence of certain components in the definitions of agility in the literature, we find that most of the authors have agreed to the original interpretation of agility in software development in the Agile Manifesto or built their view in line and on top of the original view. Majority of the Agile Manifesto authors represented advocates of different agile methods and the manifesto is a collection of these common values of agile methods. Techniques and practices in those methods can be very different, but they have a common value base and principles. We believe this also explains the acceptance of Agile Manifesto as the baseline for defining agility in ISD and software development.

Many authors, including [1], [3], [6], [13] and [19], consider agility as a form of capability or as a set of capabilities. Aligned with the definition by Lyytinen and Rose [6], we find agility as an organizational capability to respond to changes in the environment and to refine this ability. In the context of software firms, we also agree with the view of agility presented by Lee and Xia [19]. Their definition reflects more specific software development capabilities that enable the firm to produce software quickly to match the needs to the customers and the market. Overall, agility seems to express the capabilities of the management of firms engaging in software development.

Table 1. Components of agility in the extant literature.

Article	Agile Manifesto (2001)	Conboy's definition (2009)	Iterations, increments, continuous publishing	Flexibility	Quick, fast, nimble	Leanness, parsimony	Respond to change, adaptive	Face-to-face / continuous communication	Customer collaboration
Abrahamsson, Salo, Ronkainen & Warsta, 2002 [11]	x		x		x	x	x	x	x
Cockburn & Williams, 2003 [13]	x		x				x		
Ceschi, De Panfilis, Sillitti & Succi, 2005 [14]						x	x		x
Merisalo-Rantanen, Rossi & Tuunanen, 2005 [15]	x		x	x				x	x
Erickson, Lyytinen & Siau, 2005 [7]	x			x	x	x	x		
France, Rumpe & Turk 2005 [16]	x		x		x		x	x	x
Lyytinen & Rose 2006 [6]			x		x		x		
Mahapatra, Mangalaraj & Nerur, 2005 [10]	x		x	x	x	x	x	x	x
Balijepally & Nerur, 2007 [17]			x	x	x		x	x	x
Dybå & Dingsøy 2008 [5]	x		x	x	x	x	x		
Abrahamsson, Haikara, Pikkarainen, Salo, & Still, 2008 [9]	x		x				x	x	
Balijepally, Mahapatra, Nerur & Price, 2009 [18]	x		x				x	x	x
Conboy, 2009 [3]				x	x	x	x		
Lee & Xia, 2010 [19]	x		x		x		x	x	x
Baskerville, Madsen & Pries-Heje, 2011 [12]	x	x			x		x		
Cusumano & Poppendieck, 2012 [20]	x		x	x		x			x
Balijepally, Dingsøy, Moe & Nerur, 2012 [1]	x	x	x	x	x	x	x		x
Conboy, Lang & McHugh, 2012 [21]	x		x					x	
van Vliet & van Waandenburg, 2013 [22]		x			x		x		x
Petter & Yu, 2014 [23]		x	x				x	x	x

3. Measuring agility

Balijepally, Dingsøy, Moe and Nerur [1] examined the state of research of agility done in 2001-2010. They discovered that the number of articles published increased each year, being over 1500 in total. The research on agility has used various theoretical perspectives, including positions of knowledge management, personality research, and organizational learning, but the majority of articles has not aimed to operationalize or develop theories at all [1]. Regardless of the fact that research on agility is still increasing in numbers, it is still lacking a commonly accepted definition and tested theories that explain the effects of agility [19]. If we cannot define what kind of software development is agile and what is not, we cannot make claims about its impact on customer satisfaction, performance or competitive advantage. There are, however, some earlier attempts to develop approaches to measure agility.

First and foremost, agility has been measured to determine its effect on software development. The performance of agile software development can be measured by counting the amount of lines of code, counting the functionalities of the finished software and measuring added value compared to investment [24]. Agile methods are often used as basis for measuring as was done in a research where seven Scrum related indicators were used to find out the perceived benefits of agile development [24]. Another research used different agile methods to measure the change in the teams' communication habits [9]. Concepts from project management such as customer relationship, requirement control and finances have been used to measure the project managers' ability to control the team in agile development [14]. These studies measured the agile methods used, but not the agility of the method.

Conboy [3] approached agility in information system development on a general level. His definition of agility was turned into a taxonomy, which includes prescriptive conditions for agile development and

which was used to examine the agility of two projects using Scrum and XP methods. The research showed that many of the practices in agile methods did not contribute to the agility of the development and agility could be detected in practices that were not part of the agile methodology [3]. The taxonomy and its application are intended to differentiate between a practice that truly contributes to agility and one that does not.

Lee and Xia [19] applied their view of agility in integrated approach mixing qualitative and quantitative analysis in attempt to discover dependencies between development team attributes, agility in software development and project results. They define agile software development in line with the Agile Manifesto as “the software team’s capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle” [19]. First phase of the research used preliminary interviews and other techniques to fine-tune the research setting and the measuring instrument. In the second phase, they conducted a large survey for project managers. After the survey, they conducted 17 more inter-views about 10 projects that participated in the survey to better understand the results of the survey. Based on the results the authors suggest that time, budget and results should be determined first for a project and then go backwards in the model defining how much the different aspects of agility are needed and finally how autonomic and diverse the team must be [19].

Soundararajan, Balci and Arthur [4] created a method for assessing the “goodness” of an agile method. The method assesses the adequacy, capability and effectiveness using the objectives, principles and strategies framework. The framework is based on Agile Manifesto’s principles and has five objectives, nine principles and 17 strategies and identifies linkages and relationships between them. The researchers conducted an on-site study, scored the observed methods and conducted surveys gaining three kinds of scores: computed, expert, and organization members. The results showed considerable capability of supporting agile methods, but no variation between development teams was detected and results were statistically significant only in the strategies level [4].

Olszewska et al. [25] measured agile transformation quantitatively to gain evidence of its impact, benefits and drawbacks. They created a set of eight metrics to measure transformation from plan-based methods to agile methods (Scrum and Kanban) by arranging workshops with the target company and using appropriate ones from literature. They made sure that the metrics were in line with the Agile Manifesto’s values and principles. The data was gathered from logs from customer service requests, database of trouble

reports, version control tool reports and other pre-processed data with the help of the organization members who helped to identify the appropriate data in several iterations of interviews. The most significant improvement changes in the transformation were measured in the change in throughput (developed and deployed functionality per money spent) and the number of releases in a certain time period [25].

We conclude that lack of common conceptual and theoretical basis for agility has resulted in very varied results in prior studies. Most of the existing research articles report case studies and only few of them apply theoretical lenses. Studies comparing multiple organizations in terms of agility and its impact or studies where prior research setting and approach would have been repeated and examined are practically non-existent. It is therefore difficult to generalize much about the benefits and disadvantages or impact of employing agile methods or agile principles. This is also due to the lack of valid, reliable and commonly used measures of agility.

Examining the literature, we also learn that software firms apply agile methods in very different ways; one project within the same firm may be different from another, various approaches can be select from in different methods and parts of the process and instructions are left out. Therefore, when one wants to find out whether a software firm operating in agile manner, it is more important to find out if the firm is operating according to the values of agility, rather than if the firm applies particular method.

4. Developing initial measurement instrument for agility of software firms

Our review of previous research revealed, that agility in software development has been studied in a number of ways and that there are some metrics to detect its presence in software development. However, researchers define agility in different ways, which makes it hard to gain any wider understanding about agility beyond specific software development activities. In addition, we have very little knowledge about the differences between companies in relation to agility. Good measures and indicators have not been introduced, and the view on overall agility of software firms is missing altogether.

The purpose of the present study is to find out which concepts can be used to develop measurement indicators of agility in software companies and, thus, how can one measure the overall agility of a software firm. Accordingly, the study initiates work on creating measurement instrument for evaluating overall agility

of a software firm. We perceive agility as a capability of software firms that is associated with decisions to employ agile principles and methods. Therefore the purpose was to develop reflective, rather than formative measurements. Churchill [26] has presented a method, which has been deemed suitable for developing reflective measures. It comes with eight phases to create a measure, which we use here to develop of the measuring instrument. The phases include 1) specifying the domain, 2) generating sample of items, 3) collecting data, 4) purifying measure, 5) collecting data, 6) assessing reliability, 7) assessing validity and 8) developing norms. This sequence of steps was considered as simple and practical starting point especially for the development of initial measure and items and as easily extendable with more contemporary guidelines for scale development. In practice, the phases constitute an iterative process of generating and validating the measuring instruments. In this paper, we report the results of the first iteration of the work, which consists of the first four step of the Churchill's method and results in purified measurement instrument. In addition to following to Churchill's process, we considered the guidelines and

detailed advice for scale development by Straub [27], DeVellis [28] and MacKenzie, Podsakoff and Podsakoff [29].

4.1. Generating new measurement items

The first step in Churchill's [26] process involves determining the higher level concepts. In the identified literature, the concept of agility refers to both use of agile methods, agile principles and capabilities. We understand agility of a software firm as its organizational capability to respond to external changes and as its software development capability to produce functional software fast. We also believe these capabilities are the consequences of software firm's capabilities to embrace and implement agile principles. This is to say, in order to achieve organizational capabilities, the individual developers and development teams working for the firm need to be skilled and capable to adopt and make use of the agile practices.

For the purpose of studying agility in software firms, and to explore the possibilities to develop a reflective measure for software firm's agility, we opted

Table 2. Measurement items for evaluating agility.

The working methods of our development teams are organized...		
B 1	...so that creating documents is an essential part of their duties.	...so that creating documents is a minor part of their duties.
A 1	...so that they are guided by the methods and goals agreed by the team.	...using the company's methods and the goals set by the management as guidelines.
A software development project is likely to...		
C 1	...fail when there has been too much focus on the plans and documentation instead of reacting to changes.	...fail when there has been too many changes instead of keeping to the plans and documentation.
B 2	...succeed, because in addition to a working software it has produced comprehensive documentation.	... succeed, because in addition to a working software it has produced a minimal amount of documentation.
For our software development projects to be profitable, it is important that the project...		
C 2	... can deliver the order according to the customer's original wishes.	...can respond to the customer's changing requirements during the development process.
B 3	... focuses solely to produce the software.	...produces a working software as well as high-quality documentation.
The employees in the development teams...		
A 2	...are selected flexibly to different tasks according to their individual abilities.	...are selected to their tasks according to their employee profiles by their managers.
A 3	...work more productively when they form their own working methods in collaboration with their team.	...work more productively when they use the common methods adopted by the company.
It is characteristic for our projects...		
D 1	...to be more profitable, when we alter our plans flexibly according to changing requirements during the development.	...to be more profitable, when we create functioning plans in the beginning of the project and follow them.
D 2	... that the project's results are agreed in general leaving room for changes which may be later agreed on with the customer.	... that the project's results are agreed in detail before the initiation of the project.

for the main components of the Agile Manifesto and the definitions in the analyzed articles. They informed us about four higher level concepts, which together form the multidimensional concept of agility. Accordingly we assumed that agility in software firms signify the ability to effectively interact and agree on changes, the means to fully focus on producing functioning software, the ability to effectively collaborate with the customer, and the ability to promptly respond to the changes in customer demand. These capabilities highlight the four values of the manifesto and also incorporate the essential activities of software firms, namely, producing software and serving their customers.

The second step in the procedure for creating new measurement instruments is to generate items capturing the essential domain concepts. Our review of the literature indicated that despite missing a consensus between the scholars about the term, the perception of agility agreeably reflect by the four values of the Agile Manifesto. These values were chosen as basis for the development of the measurement. We aimed at four measurement scales and therefore developed 10 items in form of statements reflecting the capabilities to perform software activities according to the agile principles. These statements represented the four values of the Agile Manifesto; A) individuals and interactions over processes and tools, B) working software over comprehensive documentation, C) customer collaboration over contract negotiation, and D) responding to change over following a plan. The statements used the same vocabulary as in the Agile Manifesto wherever possible and we also sought to take into account the efficiency and effectiveness perspectives relevant to both software development activities and running software business operations.

After writing down the initial items, the content validity of the items was determined, as advised in [28], by discussing the items in a focus group of domain experts consisting of scholars, software developers and software entrepreneurs (n=7). Churchill calls such approach as “experience surveys” with “judgment sampling”. Expert review revealed how relevant the items are to the intended measurement instrument. Evaluations of clarity and conciseness from the experts were taken into consideration and causing changes to some of the items. The resulting measurement items and their operationalization in a questionnaire format are presented in Table 2.

Relation of the items to the values of the manifesto and the capabilities are marked from A to D. In the questionnaire, the statements had options for both agile and plan-based operations. That is, every statement had a pair of alternative answers with an associated five point likert scale. The scale had a neutral choice in the

middle and it was possible to leave the question unanswered.

4.2. Data collection and purifying the measure

According to Churchill [27], the next step is to gather a sample of test answers to verify that the questions reflect the high-level concepts. Data used to test the measurement instrument were obtained from the annual Finnish Software Industry Survey targeting all software companies in Finland. The definition of software company and thus the framing of the study followed the tradition of the Software Industry Survey, focusing on firms whose main activities are providing software as either products or services to their customers. The survey was sent to a total of 1998 software firms. The data collection resulted in exactly 500 responses. After excluding the missing and partial answers, which could not be used for the purposes of this study, a total of 224 answers remained. The high number of missing answers is due to the inactive firms; the sampling frame consists of all software companies and also inactive firms are advised to return blank questionnaire form. For the purpose of exploring the measurement items and for developing the initial measurement, the number of answers from software firms of all sizes and ages was considered sufficient.

A factor analysis was used to identify the underlying factors and structure of measurement instrument for agility, and to assess the construct validity of the measure [28]. Specifically, an exploratory factor analysis was conducted to purify the measurement instrument and to identify whether the items were loading on an appropriate high-level construct [26], [27]. The data from the 224 responses were examined using Principal Axis Factoring and Oblimin as a method of rotation. Principal Axis Factoring was used as it is less sensitive to the violations of the assumption of normality and because of its descriptive nature. Oblimin was selected for rotation as it is suitable for analyses where researchers have prior assumptions of the structure.

As summarized in Table 3, the factor analysis uncovered a solution of three factors with eigenvalues greater than 1 that cumulatively accounted for 61.2% of the total variance. These three factors were labeled as effective interaction (items in group A), producing software (group B), and responding to changes (merged groups C and D). Reliability or stability of results can be evaluated by assessing the internal consistency of measurement items using Cronbach's α . The 10-item instrument had an overall reliability of 0.771 and the reliability of each factor was as follows: effective interaction = 0.653; producing software = 0.717; and responding to changes = 0.750.

At this point of analysis, items with high factor loading on multiple items and with factor loadings of less than 0.5 should be considered alarming, which was the case with two of the items. A commonly agreed rule for assessing internal consistency using Cronbach's α is that the score should be higher than 0.7 for acceptable consistency. Accordingly, we assessed the factors "responding to changes" and "producing software" as acceptable measures, but evaluated the reliability of the "effective interaction" factor as questionable. As a result, the factor and the measurement items A1-A3 were excluded from the purified measurement instrument and items C1, C2, D1 and D2 were merged into one factor.

Table 3. Factor analysis results.

Factors	C & D	B	A
Respond to changes			
D1	0,821	-0,094	-0,036
D2	0,671	0,031	0,109
C1	0,582	0,128	-0,007
C2	0,488	0,028	0,003
Producing software			
B2	0,24	0,778	0,014
B3	-0,02	0,636	-0,091
B1	0,095	0,592	0,139
Effective interaction			
A3	-0,078	0,104	0,784
A2	0,032	-0,047	0,633
A1	0,078	-0,057	0,447
Cronbach's α	0,75	0,717	0,653
Eigenvalue	3,386	1,69	1,042
Variance explained	33,90%	50,80%	61,20%

4.3. Evaluating the new measurement items

The properties and quality of a measurement scale can be assessed by evaluation of the scale's reliability, construct validity and content validity. Here, reliability has been evaluated by means of the Cronbach's alpha coefficient. Focusing on the two factors "respond to changes" and "producing software", the Cronbach's alpha coefficients are satisfactory, both above the common threshold value of 0.7. Our purified measurement instrument therefore includes two sufficiently reliable measurements. It is however obvious already from the factor analysis results in Table 3 that the measurement instrument does not exhibit satisfactory construct validity in terms of

discriminant and convergent validity. This means further development of the purified measurement instrument shall require iteration and collecting new data [26]. This will also provide an opportunity to improve the wording of the items.

This is why we underscore the consideration of content validity in the present study. There are no prior studies of the software firm's agility and therefore achieving content validity requires systematic selections of the items to be included in the measurement instrument. In this regard, we aimed at increasing the content validity by following the established guidelines of [26], [27], [28] and [29].

Generation of measurement items also need to be grounded in the literature. After conducting the first iteration of data collection and purification, we find that the capability to "respond to changes" in software firm's external environment and capability to "produce software" quickly are in line with the introduced literature in software engineering and information systems development. The former reflects the overall organizational agility, while the latter reflects the specific capabilities of software firms. Moreover, these concepts are also aligned with the term flexibility in the extant literature. Flexibility has been defined a set of management capabilities, which enables monitoring the changing business environment effectively and accelerating intended changes within the company [30],[31]. According to Gosain et al. [32], responding to changes in the environment necessitates an ability to support changes in the product or service offer, which matches with the software firm's capability to produce new software products and services. Sambamurthy, Bharadwaj and Grover [32] argue in their study that an organizational agility is a combination of customer agility, operational agility, and partnering agility. This form of agility in our view denotes an ability to respond to external changes by the means of internal operational capabilities. This view emphasizes agility as operational, rather than strategic capability, which is consistent with the definitions in the software engineering and information systems research. Overall, we have demonstrated two measures with good content validity.

Merging items in classes C and D require critical examination. While the decision to merge the items was primarily made because of the results of the exploratory factor analysis, we also find conceptual grounds supporting the decision. Capability to react to changes in customer demand requires a firm to have capabilities for effective interaction and collaboration from the outset. The capabilities to collaborate effectively and to change the plans are in our thinking both essential to achieve a capacity to respond to external changes. The downside of the decision is that

we may lose some of the semantics, if combining the two aspects of agility into one factor.

In addition to assessing the agility of software firms, the measurement instrument can also be used to compare software firms with specific dimensions. Table 4 presents the scores for the test data among Finnish software firms; frequencies of the seven remaining items of the measuring instrument. The table indicates that the measurement items are able to differentiate firms with agile capabilities from the firms with less agile capabilities.

In a commercial study conducted in the United States and Europe, 88% of respondents reported that their company used agile software development methods [2]. Comparing this figure with the variation in the test data, indicating presence of both agile and less agile software firms, it is necessary to consider the reasons for the difference. The sample, research objectives and type of survey are naturally quite different and therefore the results of the surveys are not directly comparable. The scores in Table 4, however,

raise some questions as to whether the use of an agile method leads to agile operations and further contributes to overall agility or flexibility of the firm. The same controversy was found in Conboy [3], whose research results suggest that overall agility might not be associated to the use of agile methods. While our data set does not enable to examine such interrelations reliably, we argue that these early findings call attention to further investigation of the overall agility of software firms and further development of instruments to measure the agility of software business.

When interpreting the answers of the questionnaire, the respondent's view of software development must also be taken into account as the survey was targeted at the CEOs of the company. Baskerville et al. [12] and van Vliet and van Waardenburg [22] found tensions between agile software development teams and other organizations working on in plan-based mode. This finding could be affecting this study so that, for instance, a CEO considers documents and contract negotiations more important than software developers.

Table 4. Sample statistics.

Item	Likert 1	Likert 5	Frequencies	
D1	It is characteristic for our projects to be more profitable, when we create functioning plans in the beginning of the project and follow them.	It is characteristic for our projects to be more profitable, when we alter our plans flexibly according to changing requirements during the development.	1	8 (3.6%)
			2	42 (18.8%)
			3	41 (18.3%)
			4	91 (40.6%)
			5	42 (18.8%)
D2	It is characteristic for our projects that the project's results are agreed in detail before the initiation of the project.	It is characteristic for our projects that the project's results are agreed in general leaving room for changes which may be later agreed on with the customer.	1	3 (1.3%)
			2	26 (11.6%)
			3	33 (14.7%)
			4	108 (48.2%)
			5	54 (24.1%)
C1	A software development project is likely to fail when there has been too many changes instead of keeping to the plans and documentation.	A software development project is likely to fail when there has been too much focus on the plans and documentation instead of reacting to changes.	1	10 (4.5%)
			2	46 (20.5%)
			3	54 (24.1%)
			4	79 (35.3%)
			5	35 (15.6%)
C2	For our software development projects to be profitable, it is important that the project can deliver the order according to the customer's original wishes.	For our software development projects to be profitable, it is important that the project can respond to the customer's changing requirements during the development process.	1	6 (2.7%)
			2	22 (9.8%)
			3	33 (14.7%)
			4	101 (45.1%)
			5	62 (27.7%)
B2	A software development project is likely to succeed, because in addition to a working software it has produced comprehensive documentation.	A software development project is likely to succeed, because in addition to a working software it has produced a minimal amount of documentation.	1	17 (7.6%)
			2	50 (22.3%)
			3	84 (37.5%)
			4	62 (27.7%)
			5	11 (4.9%)
B1	The working methods of our development teams are organized so that creating documents is an essential part of their duties.	The working methods of our development teams are organized so that creating documents is a minor part of their duties.	1	12 (5.4%)
			2	54 (24.1%)
			3	55 (24.6%)
			4	71 (31.7%)
			5	32 (14.3%)
B3	For our software development projects to be profitable, it is important that the project produces a working software as well as high-quality documentation.	For our software development projects to be profitable, it is important that the project focuses solely to produce the software.	1	20 (8.9%)
			2	79 (35.3%)
			3	65 (29.0%)
			4	48 (21.4%)
			5	12 (5.4%)

5. Conclusions and future directions

In this article, we discussed definitions of agility and the difficult but engaging practice of measuring agility. Compared to the current literature on agility, we were interested in measuring both the overall agility of a software firm and the agility specific to firms performing software development activities. The results of the study reflect the capabilities that are present in the management of software development activities. We believe that detecting the presence of these values or ideals is the key to understanding the effect agile movement has on the software business. We analyzed the literature to find appropriate starting point for developing such measure. Our review of literature revealed that the Agile Manifesto is indeed widely accepted basis for understanding and discussing agility in software development and software business. Employing the manifesto, contemporary conceptual components and expert opinions we developed a measurement instrument. The initial measurement items were tested against empirical data and was purified to form an acceptable measurement instrument for agility of the operations in software firms.

The research question of the present article was: *How can we measure the overall agility of a software firm?* Our review of literature and reported work on developing a measurement instrument provides a partial answer. First, components of the agility across the literature and encompassed in the manifesto can be used a basis for measuring overall agility of a software firm. This is because the overall agility of a software firm signifies a capability to respond to various demands from competitive environment. Capability to produce functional software fast is the ideal protection against inevitable consequences resulting from technological progress. Capability to react effectively to fast-changing customer requirements is relevant for both product software firms and software firms providing customer-specific services. Secondly, our efforts reported in this paper to achieve a measurement instrument for the software firms' agility offer another partial, yet complementing answer to the research question. Our work indicates that it is feasible to develop acceptable measures for assessing the ability to respond to changes in customers' demand and requirements and for assessing the tendency to focus on producing functional software. These represent operational capabilities and would contribute to the overall agility of the software firm.

Before considering implications for practitioners and researchers, certain limitations of this study should be noted. Although we conducted an exploratory

factory analysis, a new iteration of measurement item development is required for creating scales with acceptable construct validity. The measurements presented in this paper are sufficiently reliable in classifying software firms into more and less agile firms, but partially fail in accurately measuring the elaborate concept. Secondly, responses to this study were obtained only from Finnish software industry, which condition are seen to correspond to the other European markets, but differ from the US market and developing markets. An issue that further studies needs to address is the wording of the questions and whether they are suitable for a questionnaire targeted for respondents in executive officer's position. This shall increase the construct validity of the measurements.

As clearly reported, in this article we have presented only the first stages of developing a measuring instrument for software companies' agility. Further development of the instrument is important to gain more reliable knowledge about the differences between software firms and the impact of employing agile approach to the performance of the firms. When the instrument is completed it is possible to find what attributes of a company are linked to agility and if agile companies more profitable, innovative or respected by their customers. This need is associated to the notion discussed in this paper that use of agile methods or parts thereof are unlikely to be the only factors contributing to the overall agility and flexibility of software firms.

References

- [1] Balijepally, V., T. Dingsøyr, N. Moe and S. Nerur, "A decade of agile methodologies: Towards explaining agile software development", *The Journal of Systems and Software*, 85, 2012, pp. 1213-1221.
- [2] VersionOne. 10th annual State of Agile Survey, <http://stateofagile.versionone.com/>, 2014.
- [3] Conboy, K. "Agility from first Principles : Reconstructing the Concept of Agility in Information Systems Development", *Information Systems Research*, 20(3), 2009, pp. 329-354.
- [4] Soundararajan, S., O. Balci and J. Arthur, "Assessing an Organization's Capability to Effectively Implement Its Selected Agile Method(s): An Objectives, Principles", *Strategies Approach, Agile Conference (AGILE)*, 2013, pp. 22-31.
- [5] Dybå, T. and T. Dingsøyr, "Empirical studies of agile software development: A systematic review", *Information and Software Technology*, 50, 2008, pp. 833-859.

- [6] Lyytinen, K. and G. Rose, "Information system development agility as organizational learning", *European Journal of Information Systems*, 15, 2006, pp. 183-199.
- [7] Erickson, J., K. Lyytinen and K. Siau, "Agile Modeling, Agile Software Development, and Extreme Programming : The state of Research", *Journal of Database Management*, 16(4), 2005, pp. 88-100.
- [8] Agile Alliance: Manifesto for agile software development, <http://agilemanifesto.org/>, 2001.
- [9] Abrahamsson, P., J. Haikara, M. Pikkarainen, O. Salo and J. Still, "The impact of agile practices on communication in software development", *Empirical Software Engineering*, 13(3), 2008, pp. 303-337.
- [10] Mahapatra, R., G. Mangalaraj and S. Nerur, "Challenges of migrating to Agile Methodologies. Communications of the ACM, 48(5), pp. 73-78. (2005).
- [11] Abrahamsson, P., O. Salo, J. Ronkainen and J. Warsta, "Agile software development methods: Review and analysis", VTT Publications 478, 2002.
- [12] Baskerville, R., S. Madsen and J. Pries-Heje, J. "Post-agility: What follows a decade of agility?", *Information and Software Technology* 53, 2011, pp. 543-555.
- [13] Cockburn, A. and L. Williams, "Agile Software development: it's about feedback and change", *IEEE Computer*, 3(5), 2003, pp. 39-43.
- [14] Ceschi, M., S. De Panfilis, A. Sillitti, and G. Succi, "Project management in plan-based and agile companies", *IEEE Software* 22(3), 2005, pp. 21-27.
- [15] Merisalo-Rantanen, H., M. Rossi and T. Tuunanen, "Is Extreme Programming Just Old Wine in New Bottles: A Comparison of Two Cases", *Journal of Database Management*, 16(4), 2005, pp. 41-61.
- [16] France, R., B. Rumpe and D. Turk, "Assumptions Underlying Agile Software-Development Processes", *Journal of Database Management*, 16(4), 2005, pp. 62-87.
- [17] Balijepally, V. and S. Nerur. "Theoretical reflections on Agile Development Methodologies", *Communications of the ACM* 50(3), 2007, pp. 79-83.
- [18] Balijepally, V., R. Mahapatra, S. Nerur and K. Price, "Are Two Heads Better than One for Software Development? The Productivity Paradox for Pair Programming", *MIS Quarterly* 33(1), 2008, pp. 91-118.
- [19] Lee, G. and W. Xia, "Toward agile : an integrated analysis of quantitative and qualitative field data on software development agility", *MIS Quarterly*, 34(1), 2010, 87-114.
- [20] Cusumano, M. and M. Poppendieck, "Lean Software Development : A Tutorial", *IEEE Software*, 29(5), 2012, pp. 26-32.
- [21] Conboy, K., M. Lang, O. McHugh, "Agile Practices: The Impact of Trust in Software Project Teams", *IEEE Software* 29(3), 2012, pp. 71-76.
- [22] van Vliet, H. and G. van Waandenburg, "When agile meets the enterprise", *Information and Software Technology* 55, 2013, pp. 2154-2171.
- [23] Petter, S. and X. Yu, "Understanding agile software development practices using shared mental models theory", *Information and Software Technology* 56, 2014, pp. 911-921.
- [24] Johansen, T., K. Kautz, and A. Uldahl, "The Perceived Impact of the Agile Development and Project Management Method Scrum on Information System and Software Development Productivity", *Australasian Journal of Information Systems*, 18(3), 2014, pp. 303-315.
- [25] Olszewska, M., J. Heidenberg, M. Weijola, K. Mikkonen and I. Porres, "Quantitatively measuring a large-scale agile transformation", *Journal of Systems and Software*, 117, 2016, pp. 258-273.
- [26] Churchill, G., "A Paradigm for Developing Better Measures of Marketing Constructs", *Journal of Marketing Research* 16 (1), 1979, pp. 64-73.
- [27] Straub, D.W. "Validating instruments in MIS research", *MIS Quarterly* 13 (2), 1989, pp. 147-169.
- [28] DeVellis, R. F. Scale development: Theory and applications. Sage publications, 2003.
- [29] MacKenzie, S., P. Podsakoff, and N. Podsakoff, "Construct measurement and Validation Procedures in MIS and Behavioral Research: Integrating New and Existing Techniques". *MIS Quarterly*, 35(2), 2011, pp. 293-334.
- [30] Adner, R. and C. Helfat, "Corporate effects and dynamic managerial capabilities", *Strategic Management Journal*, 24(10), 2003, pp. 1011-1025.
- [31] Benitez-Amado, J. and G. Ray, "Introducing IT-enabled business flexibility and IT integration in the acquirer's M&A performance equation", *ICIS 2012 Proceedings*, 2012.
- [32] Sambamurthy, V., A. Bharadwaj, and V. Grover, "Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms", *MIS Quarterly*, 2003, pp. 237-263.
- [33] Gosain, S., A. Malhotra and O. El Sawy, "Coordinating for flexibility in e-business supply chains", *Journal of Management Information Systems*, 21(3), 2004, pp. 7-45.