The Journal of Systems and Software 107 (2015) 38–49

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

The prospects of a quantitative measurement of agility: A validation study on an agile maturity model

Lucas Gren [a,*], Richard Torkar [a,b], Robert Feldt [a,b]

[a] Chalmers University of Technology and the University of Gothenburg, Gothenburg SE-412 96, Sweden
[b] Blekinge Institute of Technology, Karlskrona SE-371 79, Sweden

ARTICLE INFO

Article history:
Received 11 September 2014
Revised 18 February 2015
Accepted 1 May 2015
Available online 19 May 2015

Keywords:
Agility
Empirical study
Validation

ABSTRACT

Agile development has now become a well-known approach to collaboration in professional work life. Both researchers and practitioners want validated tools to measure agility. This study sets out to validate an agile maturity measurement model with statistical tests and empirical data. First, a pretest was conducted as a case study including a survey and focus group. Second, the main study was conducted with 45 employees from two SAP customers in the US. We used internal consistency (by a Cronbach's alpha) as the main measure for reliability and analyzed construct validity by exploratory principal factor analysis (PFA). The results suggest a new categorization of a subset of items existing in the tool and provides empirical support for these new groups of factors. However, we argue that more work is needed to reach the point where a maturity models with quantitative data can be said to validly measure agility, and even then, such a measurement still needs to include some deeper analysis with cultural and contextual items.

1. Introduction

The study of agile development and management practices is a relatively new field of research. The term itself, "agile development", was first coined in the area of software development but similar concepts preceded it in the literature on manufacturing. Today it has become a general project management concept/tool, and the word "agile" is frequently used in the general business and project management literature, e.g. Miles (2013), Poolton et al. (2006), Vinodh et al. (2010).

Agile methods in software engineering evolved during the 1990s and in 2001 it became a recognized concept due to "The manifesto for agile software development" written by a group of software developers (Fowler and Highsmith, 2001). According to Cobb (2011) the background to the agile ideas was that projects in crisis sometimes took on more flexible ways of thinking and working and then were more successful. This style was named "agile", which literally means to be able to move quickly and easily (Fowler and Highsmith, 2001), and emerged in reaction to more traditional project management methods were detailed planning typically precedes any implementation work.

During the 1990s the traditional way of doing procurement, elicitation of requirements, contract negotiations and then production and, finally, delivery (e.g. what is often termed the waterfall model in software development literature), sometimes helped create computer and software systems that were obsolete before they were delivered. To try to solve these challenges the agile community thus defined a set of values that they summarized in the agile manifesto (Fowler and Highsmith, 2001):

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

Laanti et al. (2011) claim that scientific and quantitative studies on agile methods were still rare in 2011, while requesting such studies since they can give more general advice about the practices involved. Overall, if an organization wants to transition to more agile ways of working, regardless of whether they are a software organization or not, the decision-makers will benefit from measuring agility both before, during, and after such a transition. The question is if this is possible since agility is a cultural change (described in the agile manifesto above) as well as a smorgasbord of practices to support them (Ranganath, 2011; Williams, 2012; Zieris and Salinger, 2013).

There is a diversity of agile measurement tools out there, both scientific and commercial but almost none of them has been statistically validated. In order to measure agility and trust in the given results/output, both researchers and practitioners need validated tools

* Corresponding author. Tel.: +46 739 882 010.
E-mail addresses: lucas.gren@cse.gu.se (L. Gren), richard.torkar@cse.gu.se (R. Torkar), robert.feldt@bth.se (R. Feldt).

http://dx.doi.org/10.1016/j.jss.2015.05.008
0164-1212/

to guide their process. The problem is what to focus on and on what level, since the agile approach is on a diversity of levels in the organization. This empirical study will evaluate one of the agility maturity models found in research through a statistical validation process. This tool focuses a bit more on behavior and not only lists a set of practices for the research subjects to tick yes or no regarding if they are implemented or not. We also connect a Likert scale to the evaluation in order to capture more variance in connection to each item. Section 2 will outline existing agile measurement tools found in the literature, Section 3 will present how our main statistical investigation was conducted, but also describe a pretest conducted before the main study including its findings under Section 2.2, Section 4 will present main study findings, Section 5 will analyze and discuss these overall results, and, finally, Section 6 will present conclusions and suggest future work.

This study aims to contribute with the following:

1. A test to evaluate if the agile adoption framework can be used to measure current agility (instead of agile potential).
2. If practitioners think such an evaluation is relevant through a case study pretest.
3. Expand the agile adoption framework to include a Likert scale evaluation survey filled out by all the team members and not just by the assessor/researcher and connect a confidence interval to the item results.
4. Partly validate the agile adoption framework with statistical tests.
5. Suggest changes agile adoption framework and/or highlight the issues connected to agility measurement.

2. Related work

Some researchers suggest qualitative approaches like interviewing as a method for assessing agility in teams (Boehm and Turner, 2003; Pikkarainen and Huomo, 2005; Sidky et al., 2007). Hoda et al. (2012) even suggest the use of grounded theory which is an even more iterative and domain specific analysis method (Glaser and Strauss, 2006). Interviewing is a good way to deal with interviewee misinterpretations and other related biases. The work proposed by Lee and Xia (2010) compares a few agility dimensions with performance and draw conclusions about the complexity of if agile methods increase performance or not, which they do.

Datta (2009) describes an Agility Measurement Index as an indicator for determining which method of Waterfall, Unified Software Development Process (UP), or eXtreme Programming (XP) should be used. Where Waterfall is plan-driven and XP is an agile method, UP is considered to have elements of both and is a more general framework that can be adapted to specific needs but that is often used as a kind of middle ground between the other two. The author suggests that the five dimensions: duration, risk, novelty, effort, and interaction should be taken into account when selecting development method. Their method is, however, a company-specific assessment, which makes comparisons between different organizations cumbersome.

To be able to compare and guide organization in their agile implementations a diversity of agile maturity models have been suggested, as mentioned in Section 1. Leppänen (2013) presents a useful overview of these agile maturity tools selected with the following criteria: “domain” (the domains the models are targeted to), “purpose” (the purposes the models have been developed for), “conceptual and theoretical bases” (the conceptual and theoretical backgrounds upon which the models have been built), “approaches and principles” (the approaches and principles used to construct the models), “structure” (the architectures of the models), and “use and validation” (extent of deployment and validation). Based on these criteria eight tools were selected: the agile maturity model (Ambler, 2010), a road map for implementing extreme programming (Lui and Chan, 2006), toward maturity model for extreme programming (Nawrocki et al., 2001), the agile maturity map (Packlick, 2007), agile maturity model (Patel and Ramachandran, 2009), agile maturity model (Leppänen (2013)), a framework to support the evaluation, adoption and improvement of agile methods in practice (Qumer and Henderson-Sellers, 2008), and the agile adoption framework (Sidky et al., 2007). According to Leppänen (2013) some of them are merely based on conceptual studies, others are developed only in one organization, a third group has gathered more experience from organizations, and some are discussed with practitioners. However, as also Leppänen (2013) concludes, none of them are validated. He also states that higher maturity levels could partially be assessed by more lightweight methods.

A process control method often used within IT is the American CMMI (Capability Maturity Model Integration) or the European ISO/IEC 15504 SPICE (Software Process Improvement and Capability Determination). These methods also divide the organization into different maturity levels and are essentially a set of requirements for engineering processes, particularly those involved in product development. Just like stage-gate project management these older methods often co-exist with agile methods when implemented (Turner and Jain, 2002). Since agile development processes are more of a cultural change we want to use a value-driven agile maturity model connected to measuring such behavior, i.e. we want the model we use to be built on the agile principles and not on process maturity per se.

Ozcan-Top and Demirors (2013) also compared and evaluated different agile maturity models based on fitness for purpose, completeness, definition of agile levels, objectivity, correctness, and consistency. According to their analysis Sidky’s agile adoption framework was given the best assessment results. Recently, the study by Jalali et al. (2014) showed that a set of agile measurement models give different results when tested with practitioners. This further motivates our study’s scientific validation approach to such measurements (it is obvious to us that they will not show the same results since they have not been scientifically validated).

In this study we selected to focus on the Sidky’s agile adoption framework, and in order to keep the number of items as low as possible, we selected only Level 1 of this tool. We should also mention that there is a set of commercial tools available, however, their scientific foundation is hard to assess.

We would like to highlight the difficulty of measuring something that is an ambiguous construct, such as agility. Maturity is of course even harder to assess in connection to agility since maturing with a unspecific concept is even harder. However, there are some behaviors connect to “being agile” in software development and behavior connected to this way of working, which is our definition of agile maturity in this case. We do not aim to find a way to quantitatively measure agility in this study (and we neglect the agile practices’ effectiveness/quality as well), but instead to test one of the existing tools and try to understand how to proceed in measuring/dealing agility transformations in organizations.

2.1. Sidky’s agile adoption framework

In order to determine which agile methods an organization is ready to use, Sidky (2007) suggests a method called the agile adoption framework. He motivates its use by arguing that even though there are many success stories in agile development, they are not really generalizable, i.e. it is unclear how the case by case descriptions can be used to judge agility readiness for a company which has some, but not all, aspects in common with reported cases. Sidky also criticizes more general frameworks, since they address agility in its generic form and not the actual practices.

Sidky’s approach is based on a tool that has two parts. The first part is called the agile measurement index (the same name as Datta (2009) uses, but a different tool) and is:

Table 1
Agile levels, principles, and practices (Sidky, 2007).

	Agile principles				
	Embrace change to deliver customer value	Plan and deliver software frequently	Human-centric	Technical excellence	Customer collaboration
Level 5	Low process ceremony	Technical excellence	Ideal agile physical setup	Test-driven development, paired programming, etc.	Frequent face-to-face interactions between developers and users (collocated)
Level 4	Client-driven iterations, continuous satisfaction feedback	Smaller and more frequent releases (4–8 weeks), adaptive planning		Daily progress tracking meetings, agile documentation, and user stories	Customer immediately accessible, and customer contract revolves around commitment of collaboration
Level 3		Risk-driven iterations, plan features not tasks, and maintain a backlog	Self-organizing teams, and frequent face-to-face communication	Continuous integration, continuous improvement (refactoring), unit tests, etc.	
Level 2	Evolutionary requirements	Continuous delivery, and planning at different levels		Software configuration management, tracking iteration progress, and no big design up front	Customer contract reflective of evolutionary development
Level 1	Reflect and tune process	Collaborative planning	Collaborative teams, and empowered and motivated teams	Coding standards, knowledge sharing tools, and task volunteering	Customer commitment to work with developing team

- A tool for measuring and assessing the agile potential of an organization independent of any particular agile method (based on behavior connected to practices that fit into the agile manifesto).
- A scale for identifying the agile target level will ultimately aim to achieve.
- Helpful when organizing and grouping the agile practices in a structured manner based on essential agile qualities and business values.
- Able to provide a hierarchy of measurable indicators used to determine the agility of an organization.

We only use the first part from this framework since we only want to measure behavior connected to agile practices (see Sidky, 2007 for more details on his framework).

The agile adoption framework is divided into agile levels, principles, practices and concepts, and indicators. The concept of an agile level collects a set of practices that are related and indicates the degree to which a core principle of agility is implemented. An agile principle is a set of guidelines that need to be employed to ensure that the development process is agile; the principles used are derived from the basic and common concepts of all agile methods. The agile practices and concepts are tangible activities that can be used to address a certain principle. (Table 1 shows the agile principles and their practices on the different levels.)

Sidky defines "how agile" a company is by the amount of agile practices they use. This makes a measurement tool possible and straightforward, and means that an organization that uses ten agile practices is considered to be more agile than one that uses three. The indicators are then connected to these practices and divided into respondent groups such as developers, managers and assessors, but the assessors do all the evaluations on a Likert scale from 1 (strongly disagree) to 5 (strongly agree) based on interviews. We believe the assumption that higher number of implemented practices necessarily implies more agility, is wrong since teams can use agile practices without having them aligned with the agile principles, which is also supported by research (see e.g. Zieris and Salinger, 2013). However, we still believe the items presented in the tool measures behavior connected to "agility". When it comes to investigating social processes we believe a focus on behavior instead of practices gives a better description of what happens in an organization.

Sidky sorts all practices in different agile levels depending on how "advanced" they are. We think this division of practices is arbitrary but for simplicity we have chosen to evaluate our method at a level corresponding to Level 1 to keep the number if items to a minimal. It would, of course, be advantageous to validate all levels, which we intend to do in the future. We generally do not believe a hierarchical model of practices is a good model for agility in organization. For example, why would technical excellence be on the highest level and collaborative planning on the lowest? We do not believe it makes sense to state that collaborative planning is a prerequisite for technical excellence. Table 2 shows all the agile practices assessed at Level 1. Each characteristic is evaluated through a combination of indicators taken from both developer and manager interviews. Below Table 2 you will also find a description of what the agile characteristics set out to determine.

The tool created by Sidky (2007) is based on interviews and assesses the level of agility an organization is prepared to implement and recommends what particular methods should be used. However, in order to make sure we collect the variance in the responses, we decided to measure teams that state they work with some agile methods already. The method of interviewing to assess agility is also time-consuming and it would be an advantage if this could be done as a survey instead. This is also, partly, necessary in order to use statistical analysis methods. Sidky defines agile practices and connects indicators (or items) to them according to his opinion, i.e., no statistical method was used, neither was the creation of his framework clearly based on empirical data from actual teams. He then evaluated the items by letting expert agile practitioners give their feedback on the tool. No further validation has been conducted.

This study includes two parts. First, we tested Sidky's tool on two teams at Volvo Logistics in Sweden by letting the team members fill out the survey ($N = 15$). By doing this we received many data points for each team instead of having an assessor note one data point for each. We then fed this result back to the teams in a focus group to see if they thought it was true for their team. The second step was to use a larger sample from two other companies ($N = 45$) to see if Sidky's (2007) items group in factors in the same way as he categorizes them, i.e. the next step in scale construction. If a scale is to be used a qualitative generation of items must be followed by a quantitative validation analysis (Giles, 2002). In this study, we chose internal consistency as

Table 2
Descriptions of what the different characteristics on Level 1 set out to determine (Sidky, 2007).

Agile practices	Category of assessment	Area to be assessed	Characteristic assessed	To determine
Collaborative planning	People	Management	Management style	See note 1 below table
			Buy-in	See note 2 below table
			Transparency	See note 3 below table
			Power distance	See note 4 below table
		Developers	Buy-in	See note 5 below table
	Project management	Planning	Existence	See note 6 below table
Collaborative team	Project management	Developers	Interaction	See note 7 below table
			Collectivism	See note 8 below table
			Buy-in	See notes 9 & 10 below table
Standards (coding)	People	Developers	Buy-in	See note 11 below table
Knowledge sharing	People	Developers	Buy-in	See note 12 below table
		Managers	Buy-in	See note 13 below table
Task volunteering (not task assignment)	People	Management	Buy-in	See note 14 below table
		Developers	Buy-in	See note 15 below table
Empowered and motivated teams	People	Developers	Decision making	See note 16 below table
			Motivation	See note 17 below table
			Trust	See note 18 below table
Reflect and tune process	People	Developers	Buy-in	See note 19 below table
		Managers	Buy-in	See note 20 below table
	Process	Process improvement	Capability	See note 21 below table

1. Whether or not a collaborative or a command–control relation exists between managers and subordinates. The management style is an indication of whether or not management trusts the developers and vice versa.
2. Whether or not management is supportive of or resistive to having a collaborative environment.
3. Whether or not management can be open with customers and developers, i.e., no politics and secrets.
4. Whether or not people are intimidated/afraid to give honest feedback and participation in the presence of their managers.
5. Whether or not the developers are willing to plan in a collaborative environment.
6. Whether or not the organization does basic planning for its projects.
7. Whether or not any levels of interaction exist between people thus laying a foundation for more team work.
8. Whether or not people believe in group work and helping others or are just concerned about themselves.
9. Whether or not people are willing to work in teams.
10. Whether or not people recognize that their input is valuable in group work.
11. Whether or not the developers see the benefit and are willing to apply coding standards.
12. Whether or not developers believe in and can see the benefits of having project information communicated to the whole team.
13. Whether or not managers believe in and can see the benefits of having project information communicated to the whole team.
14. Whether or not management will be willing to buy into and can see benefits from employees volunteering for tasks instead of being assigned.
15. Whether or not developers are willing to see the benefits from volunteering for tasks.
16. Whether or not management empowers teams with decision making authority.
17. Whether or not people are treated in a way that motivates them.
18. Whether or not managers trust and believe in the technical team in order to truly empower them.
19. Whether or not developers are willing to commit to reflecting about and tuning the process after each iteration or release.
20. Whether or not management is willing to commit to reflecting about and tuning the process after each iteration or release.
21. Whether or not the organization can handle process change in the middle of the project.

the main measure for reliability and analyzed construct validity by exploratory factor analysis.

Next we will present a pretest conducted with two teams at Volvo Logistics. This part of the study tests a survey approach to Sidkys tool on a small sample ($N = 15$). The purpose was to evaluate the results with the teams afterward in order to assess the appropriateness of using the tool in this manner. After this assessment we present the main methodology of the study in Section 3. We then proceed and use the tool on a large sample ($N = 45$) and conduct statistical validation tests, which is in focus for the rest of this paper. Fig. 1 shows the methodology used throughout the entire paper.

2.2. Pretest

Since the pretest aims to analyze the use of a survey tool by conducting a focus group, it comprises of two research methodologies: (*i*) a descriptive survey with the purpose of gathering quantitative data and, (*ii*) an exploratory case study with the purpose of gathering qualitative data. We ultimately believe that by using these two methods we will be able to indicate if we can collect quantitative data from the team members using the agile adoption framework.

2.2.1. Pretest case and subjects selection

The teams used in this pretest, were two teams with the same manager (Scrum Master) at Volvo Logistics[1] in Sweden. Volvo

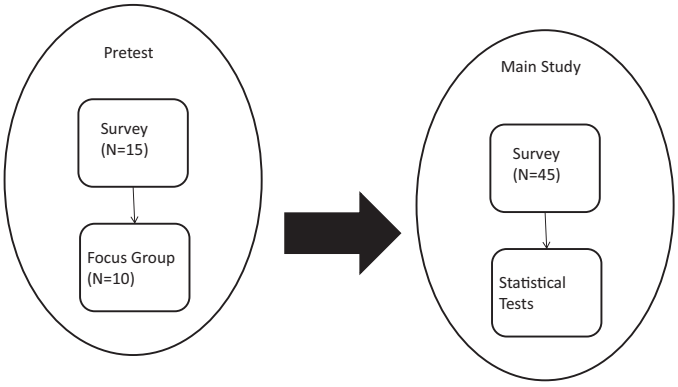

Fig. 1. Overview of the methodology used.

Logistics is a part of the Volvo Group which provides world-wide supply chain expertise to a set of automotive companies. The IT part is, of course, essential for the company to function. Many organizations, independent of field, need an efficient IT department to provide good solutions for the whole organization. The organization decided to work with agile methods and were conducting a pilot study in order to later spread the methods to other departments of the organization.

The specific teams' task was to develop a part of an enterprise software system for supply chain management. During the process they

[1] http://www.volvologistics.com

Table 3
Indicators for "collaborative planning—management style".

	N	Mean	Std. deviation
OR1_M1	7	4.71	.488
OR1_M2	7	3.71	.756
OR1_M3	6	4.00	.632
OR1_M4	7	4.43	.535
OR1_M5	7	4.57	.535
OR1_M14	7	4.14	.690
OR1_M17	7	4.14	.690
OR1_D1	8	4.25	.463
OR1_D2	8	4.25	1.165
OR1_D3	8	4.38	.518
OR1_D4	8	4.00	.756

worked with agile methods, and specifically Scrum. The reason why the sample is from software engineering is that they have the most experience with agile methods and were easier to find. The project was divided into two teams with the same manager (Scrum Master) consisting of a mixture of business- and programming-focused employees. This was done in order to assert the business effects of the project and create a method that more people could use within the organization. This meant, also, that many of the team members had managerial tasks during the project. Since there were unclear lines drawn between the teams and they had the same manager (Scrum Master), we chose to analyze the data collectively for both teams.

2.2.2. Pretest data collection procedures

Data were collected via a paper survey with items connected to agile principles for Level 1 of Sidky's (2007) tool (see Table 1). As this table shows, Level 1 is a set of practices that is defined as the first level of agility in the tool.

Instead of conducting interviews with all the team members they filled out the indicators themselves in the survey on a Likert scale from one to five and the assessor observational indicators were left out. Since Sidky's (2007)) tool has indicators on behavior connected to working with agile practices it is suitable to let the team members fill out the evaluation themselves instead of having one person do the assessment after an interview. The other studies that aim to measure agility simply state an agile principle, which forces the assessor to explain these concepts so all members know how to assess them (thus introducing the risk of bias). This also makes it possible to statistically create a confidence interval for the result based on the *t*-distribution as descriptive statistics, since a sample of many individuals is collected instead of just one. This, also, captures the deviation from the mean and the result for an indicator can then be given with a probability as confidence interval (see next section for a more thorough explanation of the procedure).

The survey was handed out in paper form to 23 team members in the two teams and 15 filled them out. The surveys were filled out at the workplace and were anonymous. The teams had many members with managerial tasks, which make the manager sample size ($N = 7$) almost equally large as the one for developers ($N = 8$). The level of agility is, in this case, a combined level for the individuals that responded to the survey. After the survey results were summarized a focus group was conducted with 10 of the individuals that had filled out the surveys. In the focus group, the participants discussed the results and gave their opinions on its relevance. These points were written down and summarized.

2.2.3. Pretest analysis procedures

Unlike Sidky (2007) all the mean values from the surveys for each individuals were calculated for each item and then, the mean value of all indicators needed for a characteristic (e.g. "collaborative planning—management style") were transformed into a percentage with a 95% confidence interval (also reported as a percentage).

Table 4
Summarized data for the characteristic "collaborative planning—management style"; the confidence interval was calculated from a *t*-distribution with $df = 7$.

		Statistic	Std. error
Total mean		4.2403	.09643
95% confidence Interval for mean	Lower bound	4.0123	
	Upper bound	4.4684	

Table 5
Descriptive statistics for the survey for developers.

	N	Mean	Std. deviation
OR1_D1	8	4.25	.463
OR1_D2	8	4.25	1.165
OR1_D3	8	4.38	.518
OR1_D4	8	4.00	.756
OR1_D5	8	4.50	.756
OR1_D6	8	4.38	.518
OR1_D7	8	4.13	.991
OR1_D8	8	4.13	1.126
OR1_D9	8	2.88	.835
OR1_D10	8	3.63	.916
OR1_D11	8	4.38	.744
OR1_D12	8	3.87	.354
OR1_D13	8	4.38	.518
OR1_D14	8	3.88	.835
OR1_D15	8	4.25	.463
OR1_D16	8	4.25	1.035
OR1_D17	8	3.88	.354
OR1_D18	8	5.00	.000
OR1_D19	8	4.38	.744
OR1_D20	8	3.13	.835
OR1_D21	8	4.62	.518
OR1_D22	8	4.38	.518
OR1_D23	8	4.00	.756
OR1_D24	8	4.00	.756
OR1_D25	8	4.50	.756
OR1_D26	8	4.50	.535
OR1_D27	8	4.25	.886
OR1_D28	8	3.88	.835
OR1_D29	8	4.38	.518

To clarify, for example if 10 people responded to all the items included in the evaluation of "collaborative planning—manager buy-in" a mean was calculated for each of these items. In order to then assess the whole characteristic the new mean value was calculated from all the mean values used in that characteristic. So all the mean values from Table 3 were used to get the total mean in Table 4. The standard deviations were of course used to get the confidence interval for the new mean value. To get the table in Table 7, the lower, upper, and mean values were divided by five (the maximum score) so they could be presented as a percentage.

When the results were summarized, the focus group was used in order to evaluate how well the results fit reality according to the team members and the managers. This focus group was a subset of the people (10 individuals, both managers and developers) that had filled out the surveys. As mentioned before, a total of 15 individuals responded to the survey (of 23) which gives a response rate of 65%.

2.2.4. Pretest results and analysis

Summary from the surveys. The results from the eight people replying to the survey for developers (29 items) is shown in Table 5, and results from the seven people replying to the survey for managers (26 items) is shown in Table 6. One manager did not reply to two items (we have not investigated the reasons for this further).

Table 6
Descriptive statistics for the survey for managers.

	N	Mean	Std. deviation
OR1_M1	7	4.71	.488
OR1_M2	7	3.71	.756
OR1_M3	6	4.00	.632
OR1_M4	7	4.43	.535
OR1_M5	7	4.57	.535
OR1_M6	6	4.17	.408
OR1_M7	7	3.57	.787
OR1_M8	7	4.29	.488
OR1_M9	7	4.57	.535
OR1_M10	7	4.14	.690
OR1_M11	7	3.71	.951
OR1_M12	7	4.29	.488
OR1_M13	7	3.29	1.254
OR1_M14	7	4.14	.690
OR1_M15	7	4.00	.577
OR1_M16	7	3.43	1.272
OR1_M17	7	4.14	.690
OR1_M18	7	3.29	1.113
OR1_M19	7	4.29	.756
OR1_M20	7	4.86	.378
OR1_M21	7	4.43	.535
OR1_M22	7	2.29	.488
OR1_M23	7	4.57	.535
OR1_M24	7	4.14	.690
OR1_M25	7	3.86	1.215
OR1_M26	7	4.00	1.000

In order to get the interval to compare to nominal scores, the indicators belonging to each assessment category were calculated according to the previously described procedure, with one alteration to the tool. The alteration was based on the result of the items: OR1_D9 and OR1_M11 (other peoples' titles and positions intimidate people in the organization). The results from these indicators were inverted, since the aspect of intimidation of titles must be seen as an unfortunate thing when working in agile manner. It is also stated by Sidky (2007) that this item is used to determine: "whether or not people are intimidated/afraid to give honest feedback and participation in the presence of their managers", which provides further indication that the scale should be inverted. This was also later confirmed by Sidky in email correspondence. The results of all the agile practices on Level 1 are presented in Table 7.

We also did a *t*-test to see if there were any differences between how managers and developers assessed the agility level. We found no such difference ($t_7 = -.701$, $p = .495$). The reason why we did not conduct a non-parametric test was that, since the *t*-test showed no difference, neither would such a test since they are more restrictive.

Summary from the focus group. The results were shown to the focus group and the group agreed on most results. The Scrum Master was a bit concerned that the result tended to be higher than his own expectations of the teams, but the focus group expressed that they were able to respond honestly and had done so on all items. After discussing this the Scrum Master agreed and revoked this comment. The questions about planning came up and according to Sidky (2007) the items are to determine if basic planning exists. When measuring the agility of a team that tries to work agile, all members were confused if planning was good or bad. They learned to be more flexible and filled out these questions in a very different way. The focus group agreed that the questions should be altered to include "deliverables" instead of "planning". This would most likely solve the confusion regarding project planning.

Another result that was low ranked was "task volunteering" for the developers. The tool caught the confusion they had whether they could volunteer for tasks or not. This was because of the team consisted of both a business- and a development-focused employees, i.e., they had different roles and did not want to take tasks belonging to someone else.

As can be seen in Table 7 the teams that were investigated had high results on most aspects of the surveys. This could simply be due to the fact that the teams were functioning well seen from an agile perspective. We also only used the first level of Sidky's (2007) tool, which could also explain the high scores. Where there were some issues, the tool caught these aspects in the variance of the result. Since this would not have shown in Sidky's tool, this motivates letting the team fill out the surveys themselves and hence collect variance in the replies and then investigate this further.

The aspects discussed in the focus group show that Sidky's (2007) agile adoption framework is suitable for measuring current agility in

Table 7
Results for the studied teams.

Agile practices	Category of assessment	Area to be assessed	Characteristic assessed	Confidence interval (95%)	Mean value	Degree of achievement
Collaborative planning	People	Management	Management style	80–89%	85%	Fully achieved
			Buy-in	80–94%	87%	Fully achieved
			Transparency	67–86%	77%	Largely achieved
		Developers	Power distance	67–87%	77%	Largely achieved
			Buy-in	77–100%	90%	Fully achieved
	Project management	Planning	Existence	47–88%	67%	Largely achieved
Collaborative team	Project management	Developers	Interaction	83–94%	89%	Fully achieved
			Collectivism	68–100%	85%	Fully achieved
			Buy-in	75–91%	83%	Largely achieved
Standards (coding)	People	Developers	Buy-in	82–98%	90%	Fully achieved
Knowledge sharing	People	Developers	Buy-in	84–98%	91%	Fully achieved
		Managers	Buy-in	73–81%	77%	Largely achieved
Task volunteering (not task assignment)	People	Management	Buy-in	74–92%	83%	Largely achieved
		Developers	Buy-in	57–88%	73%	Largely achieved
Empowered and motivated teams	People	Developers	Decision Making	73–86%	80%	Largely achieved
			Motivation	74–93%	83%	Largely achieved
			Trust	75–90%	83%	Largely achieved
Reflect and tune process	People	Developers	Buy-in	81–99%	90%	Fully achieved
			Buy-in	82–100%	91%	Fully achieved
	Process	Process improvement	Capability	77–93%	85%	Fully achieved

Table 8
Suggested survey for managers.

Indicator	Statements	Scale	Comment
OR1_M1	You actively encourage interaction among your subordinates.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M2	Irrelevant of your personal preferences, you encourage team work over individual work.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M3	You usually seek your subordinates opinions before making a decision.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M4	You frequently brainstorm with your subordinates.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M5	You frequently encourage your subordinates to find creative solutions to problems.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M6	It is important for you to share project management information with your subordinates.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M7	If you are needed and unreachable, at any point in time your subordinates have enough information to update the customer about the exact status of the project.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M8	If a problem occurs that may affect the schedule or requirements of a project, you would update your client right away.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M9	Developers should aid in the planning of a project.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M10	Customers should be part of the planning of a project.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M11	Other peoples' titles and positions intimidate people in the organization.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	(Is reversed when calculating the result)
OR1_M12	You allow your subordinates to choose their own tasks for a project.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M13	Your subordinates have unregulated access to the customer.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M14	You frequently seek the input of your subordinates on technical issues.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M15	You believe that subordinates would perform better and be more effective if they were to choose their own tasks.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M16	You always create a plan for deliverables for a project.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	(Was: "plans for a software dev. project")
OR1_M17	It is important to involve other people while preparing the project plan.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M18	The project plans are documented.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	(The word "always" was removed)
OR1_M19	When you prepare a project plan, it should not include the details of the project from start to end; it should be focused on the next iteration while giving an overview of the overall work.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M20	Project information should be communicated to the whole team.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M21	There should be a mechanism for persistent knowledge sharing between team members.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M22	If there was a wiki or a blog set up for knowledge sharing, you believe people would use it.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M23	You are willing to dedicate time after each iteration/release to review how the process could be improved.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M24	You are willing to undergo a process change even if it requires some reworking of already completed work products.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M25	If there is a need for process change, that change should not be considered a burden on the team even if significant process changes have been made previously during the project.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_M26	Process change in the middle of the project should not be considered a disruption since the process change is worth the benefit it will bring.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	

project, if the suggested alterations are made. The reason for this is that the issues discussed in the focus group and in the interview were all visible in the survey, either in the form of a low score, or with large variance associated to it.

Some more items should be altered in the survey due to the fact that they can be used more generally than just within IT projects. Putting the word "coding" in brackets, makes the tool useful for non-software development organizations as well. The word "working" should also be added as extra information when the word "coding" is used as a verb.

With the result at hand, we suggested some changes to the items before we collect more data. Table 8 shows the suggested survey for managers and Table 9 shows the suggested survey for developers. Where there is a change made from the agile adoption framework, this is commented at the end of the tables.

Since we need as much data as possible to run a quantitative statistical analysis, we opted to only use the survey for developers in the exploratory factor analysis, which is the main focus of this study and presented next.

3. Method

3.1. Hypothesis testing

In this study we want to see if empirical data of the agile adoption framework's Level 1 survey for developers correspond to Sidky's (2007) categorization of agile practices and are reliable and valid according to statistical analyses.

Hypothesis. The agile adoption framework is valid according to quantitative tests for internal consistency and construct validity.

Table 9
Suggested survey for developers.

Indicator	Statements	Scale	Comment
OR1_D1	Your manager listens to your opinions regarding technical issues.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D2	Your manager does not micro-manage you or your work.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D3	Your manager encourages you to be creative and does not dictate to you what to do exactly.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D4	Your manager gives you the authority to make decisions without referring back to him/her.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D5	You participate in the planning process of the project you will work on.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D6	If your manager said or did something wrong, it is acceptable for you to correct and/or constructively criticize him/her face to face.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D7	It is acceptable for you to express disagreement with your manager(s) without fearing their retribution.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D8	In a group meeting, the customer suggested something about the product. You disagree and have a better idea; it is acceptable for you to express disagreement with your customer and suggest something better.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D9	Other peoples' titles and positions intimidate people in the organization.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	Is reversed when calculating the result
OR1_D10	You do a better job when choosing your own task on a project instead of being assigned one by your manager.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D11	You prefer working in a group.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D12	Indicate how often you work in groups.	Likert scale from 1 (never) to 5 (always)	Different scale items (same as before)
OR1_D13	When in a group, you feel that your participation is important.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D14	Your manager seeks your input on technical issues.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D15	Your team members seek your input on technical issues.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D16	When you run into technical problems, you usually ask your team members about the solution.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D17	You usually participate in the planning process of the project you are working on.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D18	Project information should be communicated to the whole team.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D19	There should be a mechanism for persistent knowledge sharing between team members.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D20	People should use a wiki or a blog for knowledge sharing.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D21	There should exist a (coding) standard for development.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	The word "coding" put in brackets
OR1_D22	If the organization has a (coding) standard, then developers should use it when working/(coding), even in crunch time.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	Adapted to work in non-IT organizations
OR1_D23	The organization values you and your expertise.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D24	Your manager has high expectations of you.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D25	You are motivated by your job.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D26	You are willing to dedicate time after each iteration/release to review how the process could be improved.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D27	You are willing to undergo a process change even if it requires some reworking of already completed work products.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D28	If there is a need for process change, that change should not be considered a burden on the team even if significant process changes have been made previously during the project.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	
OR1_D29	Process change in the middle of the project should not be considered a disruption since the process change is worth the benefit it will bring.	Likert scale from 1 (strongly disagree) to 5 (strongly agree)	

3.2. Participants

The sample of the main study consisted of 45 employees from two large multinational US-based companies with 16,000 and 26,000 employees and with revenues of US$ 4.4 billion and US$ 13.3 billion respectively. Both stated that they are using agile methods in their participating projects. One of the companies is in the retail business and the other is in the consumer packaged goods (CPG) industry. However, the groups participating in the research were IT projects within the companies. This study was conducted together with SAP AG[2] and they mediated the contacts.

3.3. Survey

The survey used in this study was the developer survey presented in the pretest. The survey for developers were put together in an

[2] http://www.sap.com

online survey containing 29 items for the team members to answer on a Likert scale from 1 to 5 (where 1 = low agreement to the statement, and 5 = high agreement). The survey used can be seen in Table 9.

3.4. Procedure

Two 30–45 min open-ended interviews were conducted with a manager at each company with an overall perspective of their journey toward working agile. The main reason for interviewing managers was to set a psychological contract and get a commitment to making sure the survey were filled in by as many employees as possible, but also, to get the project managers to believe in how the research can help them in the future, and offer to feed the result back to them with recommendations of how to get their group to develop further regarding agility.

The surveys were sent out to the employees via email by their manager. The survey was created as an online survey and the link to it was shared in the email. It was sent to 79 employees and 45 replied, e.g. a response rate of 57%. This response rate is just above average (55.6%) within social science research (Baruch, 1999). One reminder was sent via email by one of the managers (from one of the organizations). Filling out the survey took approximately 10 min and all the questions were compulsory. The actual items can be found in Table 9. However, they are named differently but can be found by subtracting 15 from each items in the survey for developers, e.g. item Agile41 is item OR1_D26.

4. Results

In this section we will present the result of statistical tests for internal consistency and construct validity. The former will be tested by a Cronbach's α and the latter by exploratory principal factor analysis (or PFA).

However, before these statistical tests we would like to highlight a problem with using the agile adoption framework to measure agility. The terms "manager" and "Scrum Master/agile coach" could be a source of confusion. Two respondents gave the open-ended feedback of "we have a PM and an agile coach. I consider their agile skills to be far apart which lead to some ambiguity when answering questions around 'manager'. " and "some of the questions on my manager are irrelevant or could be misinterpreted. My manager is not part of the IT organization." This ambiguity probably affected the responses since some of the individuals evidently have both a manager and a Scrum Master.

4.1. Factor analysis

The reason why we used an exploratory principal factor analysis (PFA) instead of a principal component analysis (PCA) is that a PCA is meant to investigate underlying variables in data (i.e. what factors explain most of the variance orthogonally). In a PFA, on the other hand, the variables are grouped if they correlate and explain much of the same variance (i.e. the factors in a scale should not correlate too much or too little if they are considered to explain and measure a construct). A factor analysis is a statistical help to find groups of variables that explain distinct constructs in data. For more details, see e.g. Fabrigar and Wegener (2012).

The first thing to do when conducting a factor analysis is to make sure the items have the preferences needed for such a method, i.e. they need to be correlated to each other in a way that they can measure the same concept. Testing the Kaiser–Meyer–Olkin measure of sampling adequacy and Bartletts test of sphericity is a way to do this. The sphericity was significant for the whole set of items, but the Kaiser–Meyer–Olkin measure of sampling adequacy was < .5, which implicates removal of items with low correlations to the rest of the items. An anti-image table was created and low-value items were removed, i.e. values with anti-image correlation < .5. After this the Kaiser–Meyer–Olkin measure of sampling adequacy was .713, which is acceptable. The pattern matrix is shown in Table 10 and was used to divide the items into new factors. The extraction was based on eigenvalues > 1, and the promax rotation was used since the items might be dependent. As Table 11 shows, the items are correlated to more factors than the one with the highest coefficient. This means that the division into factors is not evident and the items do not clearly reflect different factors of agility. However, it should be mentioned that a factor analysis with a sample size of $N = 45$ is generally considered low, but the sample size needed for factor analysis is dependent on e.g. communalities between and over-determination of factors (MacCallum et al., 1999). Communality is the joint variables' possibility to explain variance in a factor. Over-determination of factors is how many factors are included in each variable. In this case, the first factors have a good amount of variables/factor ratio, and factors

Table 10
Pattern matrix[a] for the agile items.

	Component					
	1	2	3	4	5	6
Agile41	.977		−.323			
Agile30	.726		.318			
Agile23	.572					
Agile29	.522				.340	
Agile34		.805	.347			
Agile35		.742				
Agile31	.420	.718				
Agile38		.524		.398		
Agile32			1.031			
Agile20			.985			
Agile16				1.081		
Agile18	.337			.729		
Agile25				.455	−.783	
Agile21					.774	
Agile22				.331	.600	
Agile40				−.333		.821
Agile33						.729
Agile42	.413	−.325				.467

Extraction method: principal component analysis. Rotation method: promax with Kaiser normalization.
[a] Rotation converged in eight iterations.

Table 11
Structure matrix for the agile items.

	Component					
	1	2	3	4	5	6
Agile41	.787					.303
Agile30	.781		.598	.413		
Agile29	.716		.495	.605	.445	
Agile23	.647		.460	.389		
Agile42	.641		.389	.520		.564
Agile34		.879	.462			.403
Agile35		.752				
Agile31	.368	.696				.351
Agile38		.654	.349	.539		.431
Agile20	.446		.952	.484		
Agile32	.340		.930	.429		
Agile16			.383	.906		
Agile18	.635		.534	.840		
Agile21			.420	.444	.813	
Agile22	.420		.340	.569	.698	.486
Agile25			.382	.456	−.686	
Agile40						.782
Agile33		.418				.715

Extraction method: principal component analysis. Rotation method: promax with Kaiser normalization.

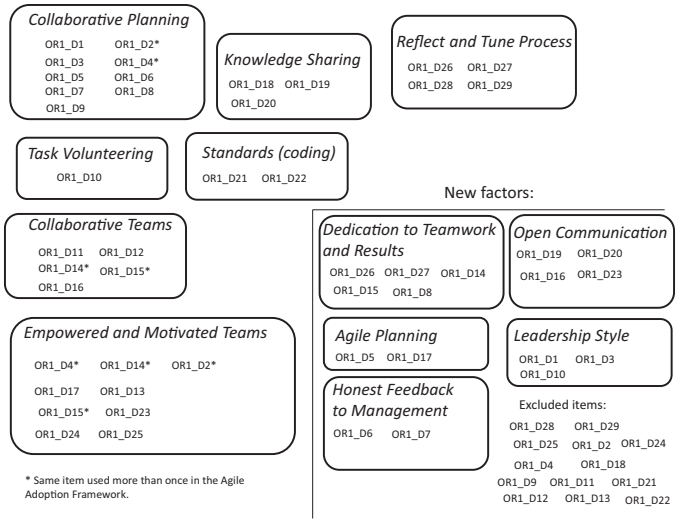

Fig. 2. Overview of which items we found support for.

3–6 include only 2 or 3 variables. The communalities are measured below with a Cronbach's α for each factor.

4.1.1. Reliability

After the new factors were created, a Cronbach's α was calculated for each new factor. The factors' α values were: .785, .761, .925, .707, .773, and .470 respectively. Values between .7 and .8 are acceptable for surveys and below .5 is unacceptable since the questions then do not cover the same construct they set out to investigate (Cronbach, 1951). The last factor (Factor 6) was therefore removed from the rest of the analysis. The other five factors were divided and named as follows: "dedication to teamwork and results" (Agile41, Agile42, Agile30, Agile23 and Agile29), "open communication" (Agile34, Agile35, Agile31 and Agile38), "agile planning" (Agile32 and Agile20), "leadership style" (Agile16, Agile18 and Agile25), and "honest feedback to management" (Agile21 and Agile22). Fig. 2 shows an overview of the items we found support for.

Since it was not possible to verify the agile practices division made by Sidky (2007) by conducting a factor analysis on data, *the hypothesis was rejected*.

5. Discussion

In this study we first tested how practitioners rate the use of the agile adoption framework through a focus group. The result of this was positive. However, the statistical tests did not support the categorization of factors in the framework and can therefore not be considered to measure distinct constructs (i.e. being a valid measurement for agility, in this case).

The pretest showed that the teams found the categories of the agile adoption framework relevant and measured how the teams worked in their new process. However, the statistical analyses suggest this measurement needs more work in order to be a valid measurement of agile practices implemented in a team. This can be due to a diversity of reasons; first, a cultural change in an organization is by definition hard to assess and very contextual. Perhaps this set of items do not reflect what agility is, however, we believe a set of items that considers a cultural as well as a behavioral dimension could be constructed in the future.

Even if the agile adoption framework does not measure the agility construct as expected and therefore the hypothesis was rejected, the items were still developed and checked for content validity by Sidky (2007), i.e. it is coherent with what some practitioners define as "agility". However, as mentioned in the introduction, a statistical analysis must support the items to be considered a valid measurement. None of the categories defined in the agile adoption framework were statistically verified. Even though this was the case, the set of items that Sidky generated are covering much of the behavior connected to agile development processes. Practitioners seem to be keen on measuring agility since they want to show proof of their success for a set of reasons, however, this does not mean the measurements really reflect agility as shown by this study.

Another possible explanation could be that our sample is too small (or skewed) to say that Sidky's categories are not supported. However, when constructing a survey tool (or "scale" in psychology) one must verify the categorizations made qualitatively through a quantitative validation. Hence, any of the mentioned agile maturity models need more development before they can be considered reliable. Furthermore, to trust the result in this study another independent PFA should be done and compared to this one. If two or more independent PFAs give the same result, we would be certain our results hold. Therefore, this result is only a first step in creating a validated tool.

Over the last decade, a diversity of agile maturity models have surfaced, as described in the introduction (Leppänen, 2013). It is a pity that researchers keep inventing new ones instead of validating (or even merging) existing tools to actually find a couple that works. Even the same year as the work of Leppänen (2013) was presented, more models have been suggested (by e.g. Soundararajan (2013)). New ideas and models are good but in this context what is really needed is to validate the existing ones so practitioners can be comfortable using them.

However, there is another fundamental issue with agile maturity models. Even if we can develop a statistically valid set of items to measure agile practices, a team's score on such a scale might not reflect what is actually meant by an agile team. The term "agile process" is undefined and many researchers and practitioners have their own definition and perception of what it exactly means. It is clear, though, that agile processes are not just a set of hands-on practices. Since agile principles are more about culture than a set of implemented methods, maybe a maturity level approach is not the way to go. Or we need to add another focus in the measurements that include cultural assessments instead of degree of used practices.

The fact that the different agile maturity models have the same agile practice in a range of different levels (Leppänen, 2013), also indicates that the maturity levels of agility are not evident. Maybe this is a syndrome of not letting go of the control mechanisms that agile principles suggest should be more in the periphery. Since agile methods are more about people and culture we suggest social psychological measurements are more appropriate if organizations want to measure their level of agility. The only study we found on social psychology and agile development processes is the article *Perceptive agile measurement: new instruments for quantitative studies in the pursuit of the social–psychological effect of agile practices* by So and Scholl (2009). Their work deserves more attention since they created a tool and validated it on a sample of $N = 227$. Since we want to measure agility in organizations, this tool will make such a measurement feasible since it excludes specific practices and focuses on behavior connected to the underlying agile principles.

The agile adoption framework is intended to assess agility before these ideas have been introduced into the organization, however, we believe an organization that has no clue what the wording "agile processes" means could still be agile in their ways of working. We also believe the opposite is true; an organization can have implemented agile practices without really being agile. Therefore, the measurement of agility should not be dependent on what the organization calls a "manager", "team lead" or "agile coach" etc., but focus on what these people are doing. This is a threat to this study since questions regarding the manager were reported to be hard to interpret. However, this is also part of our critique we just mentioned regarding building a tool that is not dependent on such jargon. The other

aspects of the tool did not form factors anyways, but we have suggested new categories for the agile adoption framework. These were: "dedication to teamwork and results", "open communication", "agile planning", "leadership style", and "honest feedback to management". This makes the agile adoption framework (Sidky, 2007) one of few agile maturity level now partially statistically validated (on Level 1 in one of the step described by Sidky). However, the questions still includes some ambiguity regarding manager and agile leader. Furthermore, the agile adoption framework uses the same items to assess both results for developers and managers, which makes statistical analysis cumbersome. However, as mentioned, in our validation we also only used the survey for developers.

Sidky's tool was not intended to measure agility of a team but agile potential. This separation of perspectives is the reason why his survey for managers does not include agile management concepts like the definition of "done". We argue, though, that a team can be agile without having implemented agile practices and therefore this type of Boolean response to if a team is agile or not before the measurement is conducted, does not cover what agility is, according to us.

We should also mention that the largest contribution by Sidky (2007), as we see it, is not his agile team level potential assessment, but the overall items regarding a go/no go decision process at an early stage to see if agile methods is a good idea for a specific organization. This part is not presented in this study but is a great contribution to the field.

We believe the work of So and Scholl (2009) could be combined with the agile adoption framework to reflect more aspects of agility in such an assessment. Then the dimensions presented in the perceptive agile measurement:

- Iteration planning
- Iterative development
- Continuous integration and testing
- Stand-up meetings
- Customer access
- Customer acceptance tests
- Retrospectives
- Collocation

can be assessed jointly with the output of this study:

- Dedication to teamwork and results
- Open communication
- Agile planning
- Leadership style
- Honest feedback to management

which we believe create a powerful and useful tool that can give teams focus points to improve. However, we believe more dimensions are still needed and can be taken from other management fields. One of these aspects that certainly affect agile adoption is, for example, to measure innovation propensity (Dobni, 2008). However, to measure all aspects of an organization in relation to agility will take time and there is always a tradeoff between doing these time-consuming expert assessment (like Sidky's entire tool) or only measuring a subset to obtain indications of focus areas, like suggested in this study.

5.1. Validity threats

Our result and therefore also our conclusions could be due to the fact that our sample is too small or that Sidky's (2007) tool is not possible to use as a quantitative tool. The ambiguity of the different perspectives (where Sidky wants to measure agile potential and we aim to measure current agility) is also a threat to validity. We have also questioned the usefulness of using these types of agile maturity models since they do not take culture, or the purpose of using agile methods, into account. Furthermore, we have used a principal factor analysis in this study which is used under the assumption that the observed variables are a linear combination of the factors. While doing this we also assume that a Likert scale generates interval data. These aspects are, however, more a part of a general discussion on the usefulness of some statistical models in social science.

6. Conclusions and future work

In conclusion, this study has shown that quantitative data do not support the categorization of a subset of items in the agile adoption framework. It is not a surprise that the categorisation made in the agile adoption framework needs more work, since no quantitative validation has been conducted. Since this is the case researchers cannot correlate quantitative agile maturity measurements to other variables in software engineering research and be confident that the results are correct. Practitioners cannot either use these tools to guide their journey toward agility. In order to create a validated survey, the items must be iterated with real data until supported and reliable. By first doing a pretest with a small sample ($N = 15$) we qualitatively validated the items. After a few alterations we ran a factor analysis and a reliability test on the tool ($N = 45$). Data did not support the division of a subset of items selected from the agile adoption framework. However, the data gave new categorizations of the items in the agile adoption framework. As far as we know, this gives one of the first partially validated agile maturity model. However, we argue that a quantitative measurement of agility as such should be complemented with cultural and contextual items to be a valid measurement of what we consider "agility" to be.

To summarize, this study has contributed with:

1. A positive result/feedback from practitioners on the usage if the agile adoption framework as measure of current agility (instead of agile potential), in a pretest case study.
2. Evolvement of the method of the agile adoption framework to include a Likert scale evaluation survey filled out by all the team members and not just by the assessor/researcher and connect confidence intervals to the item results. This way of assessing agility is less time consuming for the assessor.
3. Validation tests for internal consistency and construct validity on the agile adoption framework on additional data suggest the data collected did not support the way the indicators are related to the agile practices (on Level 1) in the framework under investigation.
4. This study finds support for a new division of items to measure agility but concludes that much validation is needed to even state that the items measure the agile practices. Furthermore, we question agile maturity models as a good way to assess agility and propose that tools look more into other dimensions like culture and innovation propensity.
5. This study also highlights the tradeoff between quick quantitative measurements to guide agile adoption that is much wanted by practitioners and time-consuming contextual and more qualitative assessments in organizations that might be closer to the real situation.

We believe the next step for this kind of research would be to combine the items from many agile maturity models and see where they overlap. These items should then be subjected to the same analysis conducted in this study with a larger data set. Obviously, the larger the sample the better when validating a tool and it would be good to validate all maturity models (including the agile adoption framework) with an even larger sample. However, we believe new separate agile maturity models have ceased to contribute to the development of measuring agility, and we want to stress the importance of creating one validated combination instead. We also see the importance of adding other dimensions than agile practices to these measurements, such as validated measurements of organizational culture and innovation propensity.

Acknowledgments

This study was conducted jointly with SAP AG[3], and we would especially like to thank Jan Musil at SAP America Inc. We would also like to thank the SAP customers who were willing to share information. Volvo Logistics, Pasi Moisander, Karin Scholes, and Kristin Boissonneau Gren (without your goodwill this work could not have been done).

References

Ambler, S., 2010. The agile maturity model (AMM). Dr. Dobbs J. April, 1.
Baruch, Y., 1999. Response rate in academic studies—a comparative analysis. Hum. Relat. 52 (4), 421–438.
Boehm, B., Turner, R., 2003. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley, Boston.
Cobb, C., 2011. Making Sense of Agile Project Management: Balancing Control and Agility. John Wiley & Sons, Inc., Hoboken.
Cronbach, L., 1951. Coefficient alpha and the internal structure of tests. Psychometrika 16 (3), 297–334.
Datta, S., 2009. Metrics and Techniques to Guide Software Development(Ph.D. thesis). Florida State University College of Arts and Sciences.
Dobni, C.B., 2008. Measuring innovation culture in organizations: the development of a generalized innovation culture construct using exploratory factor analysis. Eur. J. Innov. Manage. 11 (4), 539–559.
Fabrigar, L., Wegener, D., 2012. Exploratory Factor Analysis. Series in understanding statistics. OUP, USA.
Fowler, M., Highsmith, J., 2001. The agile manifesto, in: Software Development, Issue on Agile Methodologies, last accessed on December 29th, 2006.
Giles, D., 2002. Advanced Research Methods in Psychology. Psychology Press/Routledge, Hove, East Sussex.
Glaser, B., Strauss, A., 2006. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Transaction (a division of Transaction Publishers), New Brunswick, NJ.
Hoda, R., Noble, J., Marshall, S., 2012. Developing a grounded theory to explain the practices of self-organizing agile teams. In: Empirical Software Engineering, pp. 1–31.
Jalali, S., Wohlin, C., Angelis, L., 2014. Investigating the applicability of agility assessment surveys: a case study. J. Syst. Softw. 98, 172–190.
Laanti, M., Salo, O., Abrahamsson, P., 2011. Agile methods rapidly replacing traditional methods at Nokia: a survey of opinions on agile transformation. Inf. Softw. Technol. 53 (3), 276–290.
Lee, G., Xia, W., 2010. Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. MIS Q. 34 (1), 87.
Leppänen, M., 2013. A comparative analysis of agile maturity models. In: Information Systems Development. Springer, pp. 329–343.
Lui, K.M., Chan, K.C., 2006. A road map for implementing extreme programming. In: Unifying the Software Process Spectrum. Springer, pp. 474–481.
MacCallum, R.C., Widaman, K.F., Zhang, S., Hong, S., 1999. Sample size in factor analysis. Psychol. Methods 4, 84–99.
Miles, A., 2013. Agile learning: living with the speed of change. Dev. Learn. Organ. 27 (2), 20–22.
Nawrocki, J., Walter, B., Wojciechowski, A., 2001. Toward maturity model for extreme programming. In: Proceedings of the 27th Euromicro Conference, 2001. IEEE, pp. 233–239.
Ozcan-Top, O., Demirors, O., 2013. Assessment of agile maturity models: a multiple case study. In: Woronowicz, T., Rout, T., OConnor, R., Dorling, A. (Eds.), Software Process Improvement and Capability Determination. In: Communications in Computer and Information Science, 349. Springer, Berlin, Heidelberg, pp. 130–141. doi:10.1007/978-3-642-38833-0_12.
Packlick, J., 2007. The agile maturity map a goal oriented approach to agile improvement. In: Agile Conference (AGILE), 2007. IEEE, pp. 266–271.
Patel, C., Ramachandran, M., 2009. Agile maturity model (AMM): a software process improvement framework for agile software development practices. Int. J. Softw. Eng. 2 (1), 3–28.
Pikkarainen, M., Huomo, T., 2005. Agile Software Development of Embedded Systems Version: 1.0 date: 2005.04.04.
Poolton, J., Ismail, H., Reid, I., Arokiam, I., 2006. Agile marketing for the manufacturing-based SME. Market. Intell. Plann. 24 (7), 681–693.
Qumer, A., Henderson-Sellers, B., 2008. A framework to support the evaluation, adoption and improvement of agile methods in practice. J. Syst. Softw. 81 (11), 1899–1919.
Ranganath, P., 2011. Elevating teams from 'doing' agile to 'being' and 'living' agile. In: Agile Conference (AGILE), 2011, pp. 187–194. doi:10.1109/AGILE.2011.40.
Sidky, A., 2007. A Structured Approach to Adopting Agile Practices: The Agile Adoption Framework (Ph.D. thesis). Virginia Polytechnic Institute and State University.
Sidky, A., Arthur, J., Bohner, S., 2007. A disciplined approach to adopting agile practices: the agile adoption framework. Innov. Syst. Softw. Eng. 3 (3), 203–216.
So, C., Scholl, W., 2009. Perceptive agile measurement: new instruments for quantitative studies in the pursuit of the social–psychological effect of agile practices. In: Agile Processes in Software Engineering and Extreme Programming. Springer, pp. 83–93.
Soundararajan, S., 2013. Assessing Agile Methods: Investigating Adequacy, Capability, and Effectiveness (an Objectives, Principles, Strategies Approach)Ph.D. thesis. Virginia Polytechnic Institute and State University.
Turner, R., Jain, A., 2002. Agile meets CMMI: culture clash or common cause? In: Extreme Programming and Agile Methods: XP/Agile Universe 2002. Springer, pp. 153–165.
Vinodh, S., Devadasan, S., Vasudeva Reddy, B., Ravichand, K., 2010. Agility index measurement using multi-grade fuzzy approach integrated in a 20 criteria agile model. Int. J. Prod. Res. 48 (23), 7159–7176.
Williams, L., 2012. What agile teams think of agile principles. Commun. ACM 55 (4), 71–76.
Zieris, F., Salinger, S., 2013. Doing scrum rather than being agile: a case study on actual nearshoring practices. In: 2013 IEEE 8th International Conference on Global Software Engineering (ICGSE), pp. 144–153. doi:10.1109/ICGSE.2013.26.

Lucas Gren is a Ph.D. student in software engineering at Chalmers and the University of Gothenburg, Sweden. He has M.Sc. degrees in software engineering, psychology, business administration, and industrial engineering and management. His research focus is on decision-making, psychological aspects, agile development processes, and statistical methods (all in the context of empirical software engineering).

Richard Torkar is a professor of software engineering at Chalmers and the University of Gothenburg, Sweden. His focus is on quantitative research methods in the field of software engineering. He received his Ph.D. in software engineering from Blekinge Institute of Technology, Sweden, in 2006.

Robert Feldt is a professor of software engineering at Blekinge Institute of Technology, Sweden and at Chalmers University of Technology, Sweden. He has also worked as an IT and software consultant for more than 20 years helping companies with strategic decisions and technical innovation. His research interests include human-centered software engineering, software testing, automated software engineering, requirements engineering and user experience. Most of his research is empirical and conducted in close collaboration with industry partners in Sweden and globally. He received a Ph.D. in computer engineering from Chalmers University in 2002.

[3] http://www.sap.com