Agility Measurements Mismatch: A Validation Study on Three Agile Team Assessments in Software Engineering

Konstantinos Chronis[1] and Lucas Gren[1,2(✉)]

[1] Chalmers and University of Gothenburg, 412 96 Gothenburg, Sweden
konstantinos.chronis@gmail.com, lucas.gren@cse.gu.se
[2] University of São Paulo, São Paulo 05508–090, Brazil

Abstract. Many tools have been created for measuring the agility of software teams, thus creating a saturation in the field. Three agile measurement tools were selected in order to validate whether they yield similar results. The surveys of the tools were given to teams in Company A ($N = 30$). The questions were grouped into agile practices which were checked for correlation in order to establish convergent validity. In addition, we checked whether the questions identified to be the same among the tools would be given the same replies by the respondents. We could not establish convergent validity since the correlations of the data gathered were very few and low. In addition, the questions which were identified to have the same meaning among the tools did not have the same answers from the respondents. We conclude that the area of measuring agility is still immature and more work needs to be done. Not all tools are applicable to every team but they should be selected on the basis of how a team has transitioned to agile.

Keywords: Validation · Agile measurement · Empirical study

1 Introduction

Agile and plan-driven methodologies are the two dominant approaches in the software development. Although it has been almost 20 years since the former were introduced, the companies are quite reluctant in following them [1].

Software development teams started adopting the most known agile methodologies, such as eXtreme Programming [2], Feature Driven Development (FDD), [3], Crystal [4], Scrum [5] and others. Most companies use a tailored methodology by following some of the aforementioned processes and practices which better suit their needs. Williams et al. [6] report that all XP practices are exercised rarely in their pure form, something on which Reifer [7] and Aveling [8] also agree based on the results of their surveys, which showed that it is common for organizations to partially adopt XP. The most important issue that tends to be neglected though, is how well these methodologies are adopted.

According to Escobar-Sarmiento and Linares-Vasquez [9], the agile methodologies are easier to misunderstand. The previous statement is also supported by Taromirad and Ramsin [10], who argue that the agile software development methodologies are often applied to the wrong context. Sidky [11] defines the level of agility of a company as the amount of agile practices used. Considering this statement, a group that uses pair programming and collective code ownership at a very low level is more agile than a group which uses only pair programming but in a more efficient manner.

Williams et al. [12] pose the question "*How agile is agile enough*"? According to a survey conducted by Ambysoft [13], only 65 % of the agile companies that answered met the five agile criteria posed in the survey. Poonacha and Bhattacharya [14] mentioned that the different perceptions of agile practices when they are adopted are troublesome, since even people in the same team understand them differently, according to the result of a survey [15].

Since agile methodologies become more and more popular, there is a great need for developing a tool that can measure the level of agility in the organizations that have adopted them. For over a decade, researchers have been constantly coming up with models and frameworks in an effort to provide a solution.

This case study comprises three tools which claim to measure the agility of software development teams using surveys. These tools are Perceptive Agile Measurement (PAM) [16], Team Agility Assessment (TAA) [17], Objectives Principles Strategies (OPS) [18]. The first one has been validated with a large sample of subjects, the second one is well-used by companies and the third one covers many agile practices. Since all three tools measure agility, convergent validity should be established among them to corroborate this. The surveys from the three tools were given to Company A employees to answer. The analysis of the data was performed by grouping the survey questions in accordance to to agile practices. The correlation of these practices were the indications for establishing the convergent validity. Moreover, questions identified to have the same meaning among the tools should have the same answers from the respondents. The purpose of this study is to check whether these three tools will yield similar results.

Research Questions.

1. Will PAM, TAA and OPS yield similar results?
    (i) Does convergent validity exist between the tools?
    (ii) Will the questions that are exactly the same in the tools yield the same results?

2   Case Study

Any effort to see if the selected agility measurement tools are valid in what they do, would require to apply them to real software developments teams. According to Runeson and Host [19], a case study is "a suitable research methodology for software engineering research since it studies contemporary phenomena in their natural context". As a result, a case study was selected as the most suitable means.

2.1   Subject Selection

Company *A* is a United States company which operates in the Point Of Sales (POS) area. It has four teams with mixed members of developers and testers. The teams do not follow a specific agile methodology, but rather a tailored mix of the most famous ones which suits the needs of each team. Methodology A, as we can name it, embraces the practices from the various agile methodologies, some of them to a larger and some of them to a smaller extent. The analysis process created by Koch [20] was used for identifying these methodologies. The identification of the practices was done by observing and understanding how the teams work.

2.2   Data Collection

In order to collect the data, an online survey was considered to be the best option, since it could be easily answered by each subject.

For each of the tools, four surveys were created (one for each team). The data collection lasted about one month, while the surveys for each tool were conducted every ten days. None of the subjects was familiar with any of the tools.

Two subjects were requested to answer to the surveys first, in order to detect if there were any questions which could cause confusion, but also to see how much time is needed to complete a survey. Once the issues pointed out by the two subjects were fixed, the surveys were sent to the rest of the company's employees.

The links for the surveys were sent to the subjects via email, and they were asked to spend 15–20 min to reply to the survey. The employees who belonged to more than one team were asked a couple of days later to take the other survey in order to verify that their answers matched in both surveys.

OPS agility measurements are based on three aspects: Adequacy, Capability and Effectiveness. Effectiveness measurement focuses on how well a team implements agile methodologies. Since the rest of the tools focus on the same thing, it was decided only to use the survey from Effectiveness and not to take into account the Adequacy and Capability aspects.

The surveys for PAM, TAA and OPS were answered on a Likert scale 1–7 (never having done what is asked in the question to always doing what is asked in the question).

The employees who were asked to answer to the surveys were all members of the software development teams, which consisted of software and QA engineers. All of the participating employees have been in the company for over a year and most of them have more than five years of work experience in an agile environment. Employees who had been working for less than six months in the company were not asked to participate, since it was considered that they were not fully aware of the company's procedures or that they were not familiar enough with them. Each participant replied to 176 questions in total. Initially, 34 surveys were expected to be filled in, but in the end, 30 of them were filled in, since some employees chose not to participate.

2.3   Data Preparation

All three tools have different amount of questions and cover different practices. For this reason, we preferred to do a grouping of the questions based on the practices/areas to which they belong.

Team Agility Assessment – Areas. Team Agility Assessment (TAA) does not claim that it covers specific agile practices, but rather areas important for a team. It focuses on product ownership for Scrum teams but also on the release, iteration planning and tracking. The team factor plays a great role, as well as the development practices and the work environment. Automated testing and release planning are important here as well.

Perceptive Agile Measurement – Practices. The Perceptive Agile Measurement (PAM) tool focuses on the iterations during software development, but also on the stand-up meetings for the team members, their collocation and the retrospectives they have. The access to customers and their acceptance criteria have a high importance as well. Finally, the continuous integration and the automated unit testing are considered crucial in order to be agile.

Objectives, Principles, Strategies (OPS) – Practices. Objectives, Principles, Strategies (OPS) Framework is the successor of the Objectives, Principles, Practices (OPP) Framework [21]. OPP identified 27 practices as implementations of the principles which later on were transformed into 17 strategies.

Practices Covered Among The Tools. We have abstracted some of the OPP practices to OPS strategies in order to avoid repeating the mapping of the questions. The connection between the practices and the strategies is done based on the questions of each tool.

Mapping of questions among tools. PAM has its questions divided on the basis of agile practices, while on the other hand, TAA has divided them based on areas considered important. Although all practices/areas from PAM and TAA are mapped onto OPP and OPS, not all of their questions are under OPP practices or OPS strategies. This can be explained due to the different perception/angle that the creators of the tools have and what is considered important for an organization/team to be agile.

2.4   Data Analysis

The data gathered from the surveys were grouped on the basis of the practices covered by the OPP, and as a consequence, the OPS.

Convergent Validity Analysis. Since all the tools claim to be measuring agility and under the condition that convergent validity exists among them, then, by definition, they should yield similar results.

In similar studies [22, 23], the correlation analysis was selected as the best way to check similar tools and this was followed here as well. We decided to use the practices covered by each tool and see if they correlate with the same practices from the other two tools. The idea is based on the *multitrait-multimethod matrix*, presented by Campbell and Fiske [24]. The matrix is the most commonly used way for providing construct validity.

In order to select which correlation analysis method to choose from, the data were checked if they had normal distribution by using the Shapiro-Wilk test which is the most powerful normality test, according to a recent paper published by Razali and Wah [25]. The chosen alpha level was 0.05, as it is the most common one.

Out of the 42 normality checks (three for each of the 14 practices), only 17 concluded that the data are normally distributed. The low level of normally distributed data gave a strong indication that Spearman's rank correlation coefficient, which is more adequate for non-parametric data, was more appropriate to use, rather than the Pearson's product-moment correlation.

In order to use the Spearman's rank correlation coefficient, a monotonic relationship between two variables is required. In order to check for the monotonicity, plots were drawn between the results of each tool for all 14 practices. The plots surprisingly showed that only eight out of 42 were monotonic, which indicates no correlation what-so-ever.

Direct Match Questions Analysis. We want to find which questions are the same among the tools. In order to achieve this, the mapping described in Subsect. 2.3 was used. Afterward, the questions were checked one by one to identify the ones which had the same meaning. When we finalized the groups of questions which were the same, we requested from the same employees who were taking the pilot surveys to verify if they believed the groups were correctly formed. Their answer was affirmative, so we continued by checking if the answers of the subjects were the same. Surprisingly, OPS–TAA have 20 questions with the same meaning, while OPS–PAM and TAA–PAM only four and three respectively.

Out of the 35 normality checks (two for each group and three for one group), only 2 concluded that the data are normally distributed. Since the samples are also independent (they do not affect one another), there is a strong indication that the MannWhitney U test is appropriate. For the group *Smaller And Frequent Product Releases*, we used the Kruskal–Wallis one-way analysis of variance method, which is the respective statistical method for more than two groups.

The hypothesis in both cases was:

H_0: *There is no difference between the groups of the same questions*
H_1: *There is a difference between the groups of the same questions*

3   Results

3.1   Correlations

As it was previously stated, only eight out of 42 plots were monotonic. The more interesting than the correlations result is the non-existence of monotonicity

in the other 34 relationships, which leads us to the conclusion that there is little convergence among the tools. This is surprising because tools claiming to measure the same thing should converge.

3.2   Direct Match Questions Results

The groups of direct match questions showed some unexpected results. Questions which are considered to have the same meaning should yield the same results, which was not the case for any of the question groups, apart from one group concerning the *Software Configuration Management*. On the other hand, the *Product Backlog* practice had the lowest score with only six respondents giving the same answer. The maximum difference in answers was up to two Likert-scale points.

As far as the results from the Mann-Whitney U test and Kruskal-Wallis one-way analysis of variance are concerned, the p-values from the majority of the groups are more than the alpha level of 0.05. As a result, we cannot reject the H_0 hypothesis. Such practices are *Iteration Progress Tracking and Reporting - group #2*, *High-Bandwidth Communication* and others. On the other hand, the p-value of group *Software Configuration Management* cannot be computed, since all the answers are the same, while for other groups the p-value is below the alpha level which means that the H_0 hypothesis can be rejected. Such practices are *Continuous Integration - group #2*, *Iteration Progress Tracking and Reporting - group #4* and others.

4   Discussion

4.1   Will PAM, TAA and OPS Yield Similar Results?

The plots drawn by the data gathered showed an unexpected and interesting result. Not only do the tools lack a correlation, but they do not even have a monotonic relationship when compared to each other for the agile practices covered, resulting in absence of convergent validity. This could indicate two things; the absence of monotonicity and the negative or very low correlations show that the questions used by the tools in order to cover an agile practice do it differently as well as that PAM, TAA and OPS measure the agility of software development teams in their own unique way.

Almost all groups had different responses to the same questions. With regards to the research question ***"Does convergent validity exist among the tools?"***, we showed that convergent validity could not be established due to the low (if existing) correlations among the tools. Concerning the research question ***"Will the questions that are exactly the same among the tools yield the same results?"***, we saw that a considerable amount of respondents' answers were different.

The reasons for this somewhat unexpected results are explained in the following paragraphs.

Few or no questions for measuring a practice. A reason for not being able to calculate the correlation of the tools is that they cover slightly or even not at all some of the practices. An example of this is the *Smaller and Frequent Product Releases* practice. OPS includes four questions, while on the other hand, PAM and TAA have a single question each. Furthermore, *Appropriate Distribution of Expertise* is not covered at all by PAM. In case the single question gets a low score, this will affect how effectively the tool will measure an agile practice. On the contrary, multiple questions can better cover the practice by examining more factors that affect it.

The same practice is measured differently. Something interesting that came up during the data analysis was that although the tools cover the same practices, they do it in different ways, leading to different results. An example of this is the practice of *Refactoring*. PAM checks whether there are enough unit tests and automated system tests to allow the safe code refactoring. In case the course unit/system tests are not developed by a team, the respondents will give low scores to the question, as the team members in Company *A* did. Nevertheless, this does not mean that the team never refactors the software or does it with bad results. All teams in Company *A* choose to refactor when it adds value to the system, but the level of unit tests is very low and they exist only for specific teams. On the other hand, TAA and OPS check how often the teams refactor, among other aspects.

The same practice is measured in opposite questions. The *Continuous Integration* practice has a unique paradox among TAA, PAM and OPS. The first two tools include a question about the members of the team having synchronized to the latest code, while OPS checks for the exact opposite. According to Soundararajan [18], it is preferable for the teams not to share the same code in order to measure the practice.

Questions phrasing. Although the tools might cover the same areas for each practice, the results could differ because of how a question is structured. An example of this is the *Test Driven Development* practice. Both TAA and PAM ask about automated code coverage, while OPS just asks about the existence of code coverage. Furthermore, TAA focuses on 100 % automation, while PAM does not. Thus, if a team has code coverage but it is not automated, then the score of the respective question should be low. In case of TAA, if the code coverage is not fully automated, its score should be even lower. It is evident that the abstraction level of a question has a great impact. The more specific it is, the more a reply to it will differ, resulting in possible low scores.

Better understanding of agile concepts. In pre-post studies there is a possibility of the subjects becoming more aware of a problem in the second test due to the first test [26]. Although the *testing* threat, as it is called, does not directly apply here, the similar surveys on consecutive weeks could have enabled the respondents to take a deeper look into the agile concepts, resulting in better understanding of them, and consequently, providing different answers to the surveys' questions.

How people perceive agility. Although the concept of agility is not new, people do not seem to fully understand it, as Conboy and Wang [27] also mention. This is actually the reason behind the existence of so many tools in the field which are trying to measure how agile the teams are or the methodologies used. The teams implement agile methodologies differently and researchers create different measurement tools. There are numerous definitions of what agility is [28–31], and each of the tool creators adopt or adapt the tools to match their needs. Their only common basis is the agile manifesto and its twelve principles [32], which are (and should be considered as) a compass for the agile practitioners. Nevertheless, they are not enough and this resulted in the saturation of the field. Moreover, Conboy and Fitzgerald [33] state that the agile manifesto principles do not provide practical understanding of the concept of agility. Consequently, all the reasons behind the current survey results are driven by the way in which tool creators and tool users perceive agility.

The questions in the surveys were all based on how their creators perceived the agile concept which is quite vague, as Tsourveloudis and Valavanis [34] have pointed out. None of the Soundararajan [18], So and Scholl [16], Leffingwell [17] claimed, of course, to have created the most complete measurement tool, but still, this leads to the oxymoron that the tools created by specialists to measure the agility of software development teams actually do it differently and without providing substantial solution to the problem. On the contrary, this leads to more confusion for the agile practitioners.

Considering that the researchers and specialists in the agile field perceive the concept of agility differently, it would be naive to say that the teams do not do the same. The answers to surveys are subjective and people reply to them depending on how they understand them. Ambler [15] stated the following: "I suspect that developers and management have different criteria for what it means to be agile". This is also corroborated by the fact that, although a team works in the same room and follows the same processes for weeks, it is rather unlikely that its members will have the same understanding of what a retrospection or a releasing planning meeting means to them, a statement which is also supported by Murphy et al. [35].

5   Threats to Validity

5.1   Construct Validity

We consider that the construct validity concerning the surveys given to the subjects was already handled by the creators of the tools which were used. Our own construct validity lies in establishing the convergent validity. The small sample of subjects was the biggest threat in establishing convergent validity, making the results very specific to Company *A* itself. Future work on this topic should be performed at other companies to mitigate this threat. In order to avoid mono-method bias, some employees were asked to fill in the surveys first in order to detect any possible issues. All the subjects were promised to remain anonymous, resulting in mitigating the evaluation apprehension [36].

5.2   Internal Validity

The creators of PAM, TAA and OPS have already tried to mitigate internal validity when creating their tools. Yet, there are still some aspects of internal validity, such as selection bias maturation and testing effect. With regard to maturation, this concerns the fatigue and boredom of the respondents. Although the surveys were small in size and did not require more than 15–20 min each, still the similar and possibly repetitive questions on the topic could cause fatigue and boredom to the subjects. This could result in the participants giving random answers to the survey questions. The mitigation for this threat was to separate the surveys and conduct them during three different periods. In addition, the respondents could stop the survey at any point and continue whenever they wanted. As far as the testing effect is concerned, this threat could not be mitigated. The testing effect threat applies to pre-post design studies only, but due to the same topic of the surveys, the subjects were to some extent more aware of what questions to expect in the second and third survey. Finally, selection could also not be mitigated, since the case study focused on a specific company only.

5.3   Conclusion Validity

Although the questions of the surveys have been carefully phrased by their creators, still there may be uncertainty about them. In order to mitigate this, for each survey a pilot one was conducted to spot any questions which would be difficult to understand. In addition, the participants could ask the first author about any issue they had concerning the survey questions. Finally, the statistical tests were run only for the data that satisfied the prerequisites, with the aim to mitigate the possibility of incorrect results.

5.4   External Validity

This case study was conducted in collaboration with one company and 30 subjects only. Consequently, it is hard to generalize the outcomes. Nevertheless, we believe that any researcher replicating the case study in another organization with teams which follow the same agile practices as those used in Company *A* would get similar results.

5.5   Reliability

To enable other researchers to conduct a similar study, the steps followed have been described and the reasons for the decisions made have been explained. Furthermore, all the data exist in digital format which can be provided to anyone who wants to review them. The presentation of the findings could probably be a threat to validity because of the first author's experience at the company. In order to mitigate this, the findings were discussed with a Company *A* employee who did not participate in the case study.

6 Conclusions and Future Work

6.1 Conclusions

This paper contributes to the area of measuring the agility of software developments teams. This contribution can be useful for the research community, but mostly for practitioners. We provided some evidence that tools claiming to measure agility do not yield similar results. The expertise of the tool creators is unquestionable, but nevertheless, their perception of agility and their personal experience have led them to create a tool in the way they consider appropriate. A measurement tool which satisfies the needs of one team may not be suitable for other teams. This derives not only from the team's needs but also from the way it transitioned to agile. Companies need a tool to measure agility in order to identify their mistakes and correct them with the total purpose to produce good quality software for their customers. There is still work to be done in order to find a universal tool for measuring agility, and such a tool should be scientifically validated before it is used.

6.2 Future Work

It would be interesting to see the results of a study that would be conducted at more companies, in order to compare them to the results of the present study. In addition, another way of forming the data samples could indicate different results, which is worth looking into. Moreover, future work in the field could check for establishing convergent validity among other agility measurement tools, combine them, validate them, and finally, only use them where their output is relevant in context.

References

1. Sureshchandra, K., Shrinivasavadhani, J.: Moving from waterfall to agile. In: Agile Conference (AGILE 2008), pp. 97–101, August 2008
2. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. The XP Series. Addison-Wesley, Reading (2004)
3. Palmer, S.R., Felsing, M.: A Practical Guide to Feature-Driven Development. Pearson Education, London (2001)

 4. Cockburn, A.: Crystal Clear a Human-powered Methodology for Small Teams. Addison-Wesley Professional, Boston (2004)
 5. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Series in Agile Software Development. Prentice Hall, Englewood Cliffs (2001)
 6. Williams, L., Krebs, W., Layman, L., Antón, A., Abrahamsson, P.: Toward a framework for evaluating extreme programming. In: Empirical Assessment in Software Engineering (EASE), pp. 11–20 (2004)
 7. Reifer, D.J.: How to get the most out of extreme programming/agile methods. In: Wells, D., Williams, L. (eds.) XP 2002. LNCS, vol. 2418, pp. 185–196. Springer, Heidelberg (2002)
 8. Aveling, B.: XP lite considered harmful? In: Eckstein, J., Baumeister, H. (eds.) XP 2004. LNCS, vol. 3092, pp. 94–103. Springer, Heidelberg (2004)
 9. Escobar-Sarmiento, V., Linares-Vasquez, M.: A model for measuring agility in small and medium software development enterprises. In: 2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI), pp. 1–10, October 2012
10. Taromirad, M., Ramsin, R.: Cefam: Comprehensive evaluation framework for agile methodologies. In: 32nd Annual IEEE Software Engineering Workshopp, SEW 2008, pp. 195–204, October 2008
11. Sidky, A.: A structured approach to adopting agile practices: The agile adoption framework. Ph.D. thesis, Virginia Polytechnic Institute and State University (2007)
12. Williams, L., Rubin, K., Cohn, M.: Driving process improvement via comparative agility assessment. In: Agile Conference (AGILE 2010), pp. 3–10 (2010)
13. Ambysoft.: How agile are you? (2013)
14. Poonacha, K., Bhattacharya, S.: Towards a framework for assessing agility. In: 2012 45th Hawaii International Conference System Science (HICSS), pp. 5329–5338, January 2012
15. Ambler, S.W.: Has agile peaked? (2011)
16. So, C., Scholl, W.: Perceptive agile measurement: new instruments for quantitative studies in the pursuit of the social-psychological effect of agile practices. In: Abrahamsson, P., Marchesi, M., Maurer, F. (eds.) Agile Processes in Software Engineering and Extreme Programming. LNBIP, vol. 31, pp. 83–93. Springer, Heidelberg (2009)
17. Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. The Agile Software Development Series. Addison-Wesley Professional, Boston (2007)
18. Soundararajan, S.: Assessing Agile Methods, Investigating Adequacy, Capability and Effectiveness. Ph.D. thesis, Virginia Polytechnic Institute and State University (2013)
19. Runeson, P., Hst, M.: Guidelines for conducting and reporting case study research in software engineering. Empir. Softw. Eng. **14**(2), 131–164 (2008)
20. Koch, A.: Agile Software Development: Evaluating The Methods For Your Organization. Artech House, Incorporated, Boston (2005)
21. Soundararajan, S., Arthur, J., Balci, O.: A methodology for assessing agile software development methods. In: Agile Conference (AGILE 2012), pp. 51–54 (2012)
22. Jalali, S., Wohlin, C., Angelis, L.: Investigating the applicability of agility assessment surveys: A case study. J. Syst. Softw. **98**, 172–190 (2014)
23. Delestras, S., Roustit, M., Bedouch, P., Minoves, M., Dobremez, V., Mazet, R., Lehmann, A., Baudrant, M., Allenet, B.: Comparison between two generic questionnaires to assess satisfaction with medication in chronic diseases. PLoS ONE **8**(2), 56–67 (2013)
24. Campbell, D.T., Fiske, D.W.: Convergent and discriminant validation by the multitrait-multimethod matrix. Psychol. Bull. **56**(2), 81–105 (1959)

25. Razali, N., Wah, Y.B.: Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. J. Stat. Model. Anal. **2**(1), 21–33 (2011)
26. Campbell, D.T., Stanley, J.: Experimental and Quasi-Experimental Designs for Research. Cengage Learning, New York (1963)
27. Conboy, K., Wang, X.: Understanding agility in software development from a complex adaptive systems perspective. In: ECIS (2009)
28. Kidd, P.T.: Agile Manufacturing: Forging New Frontiers. Addison-Wesley, Reading (1994)
29. Kara, S., Kayis, B.: Manufacturing flexibility and variability: an overview. J. Manuf. Technol. Manage. **15**(6), 466–478 (2004)
30. Ramesh, G., Devadasan, S.: Literature review on the agile manufacturing criteria. J. Manuf. Technol. Manage. **18**(2), 182–201 (2007)
31. Nagel, R.N., Dove, R.: 21st Century Manufacturing Enterprise Strategy: An Industry-Led View. Diane Pub Co, Collingdale (1991)
32. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for agile software development (2001)
33. Conboy, K., Fitzgerald, B.: Toward a conceptual framework of agile methods: A study of agility in different disciplines. In: Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research, WISER 2004, pp. 37–44 (2004)
34. Tsourveloudis, N., Valavanis, K.: On the measurement of enterprise agility. J. Intell. Robot. Syst. **33**(3), 329–342 (2002)
35. Murphy, B., Bird, C., Zimmermann, T., Williams, L., Nagappan, N., Begel, A.: Have agile techniques been the silver bullet for software development at microsoft? In: 2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 75–84, October 2013
36. Wohlin, C., Ohlsson, M.C., Wessln, A., Hst, M., Runeson, P., Regnell, B.: Experimentation in Software Engineering. Springer, Berlin Heidelberg (2012)