Project Management Development – Practice and Perspectives
Fifth International Scientific Conference on Project Management in the Baltic Countries
April 14-15, 2016, Riga, University of Latvia

ADOPTION AND EVOLUTION OF AGILE PRACTICES

Marco Sampietro, SDA Bocconi School of Management, Milan, Italy
marco.sampietro@sdabocconi.it

Abstract

The traditional software product development models are characterised by a predictive approach derived from physical engineering processes. The recent developments of the software-related scenario have proved these plan-driven models to be inadequate, especially in high risk and rapidly changing environments. Agile methodologies were therefore developed as an answer to these issues. Due to their intrinsic agility, they confirmed to be capable to adapt to further changes occurred since their first introduction: methodologies were modified and tailored according to every different shortcoming found during their introduction. This research is aimed at detecting the dominant agile practices and adoption strategies, which transcends the original belonging to a certain model and may be affected by internal and external characteristics. A survey-based analysis was conducted in order to find out what are the most used practices. 194 valid questionnaires were returned. Factor analysis and cluster analysis allowed to relate practices to several environmental characteristics, with the purpose of seeking a significant relation between them. Results of the study showed that most of the companies prefer to adopt practices related to the general management of the development process, to the detriment of practices associated with coding and testing techniques. The opinions of the team members turned out to be the only inner factor responsible for the introduction or the abandonment of the practices. About the outer variables, the size of the team and the proximity of the team members have proven to significantly affect the adoption strategy.

Key words: *agile project management, agile methods, agile practices, SCRUM, Kanban, XP, Lean development*
JEL code: ***M54***

Introduction

It all started n Febraury 2001 with four values written on a website. The cornerstones of the "Manifesto for Agile Software Development" (AgileAlliance, 2001) were nothing more than four sentences, which however represented an authentic breakthrough for Project Management. Actually, the authors of the Manifesto aimed to gather a group of characteristics whose underlying values could be traced back to a unique, revolutionary way of defining a system development model. They had already created several software development methodologies (SDMs) and, after having identified the core values, decided to write a short but nevertheless inspiring list of guidelines, which should be followed by any Agile practitioner. Indeed, "rather than focusing on their differences and the competitive advantage of their own methodologies, 17 creators and supporters of the lightweight methodologies gathered (…) to discuss their common interests and philosophies, coining the term *Agile software development*" (Williams, 2012, p. 71-72). That is to say, the authors created a mind-set, a framework with general advice, which could be freely interpreted and applied in heterogeneous environments. Furthermore, the Manifesto was completed by twelve principles, which, again, could be employed regardless the particular SDM chosen for the project.

Agile is based on Values, Principles and Practices. "Agile values are the philosophy behind Agile methods, which are further defined and supported by the Agile principles and Agile practices" (Kong, Kendall, & Kendall, 2012).

Each specific project undertaken by a particular team of a certain company might be characterised by the usage of a certain set of practices, i.e. "concrete activities and practical techniques that are used to develop and manage software projects in a manner consistent with the Agile principles" (Sidky, Arthur, & Bohner, 2007).

Research questions and methodology

The importance of Agile Software Development Models (SDM) is proven to be decreasing over the recent years since agile practitioners noted that applying SDMs as is, did not generate the expected outcomes. Over the years Agile SDMs have been customised by many companies in order to find the best fit with their organisational characteristics. Therefore it is necessary to focus on the practices more than on SDMs. In addition, it would be interesting to identify the drivers that determine the adoption of certain practices. To date, however, there are no studies that address these topics.

For this reason a research study has been done in order to answer to these research questions:

1. What are the dominant Agile business practices?
2. What are the key variables that lead to a certain adoption strategy?
3. Why do the practices change within the PM groups?

In order to answer those research questions, a questionnaire-based survey has been developed and administered.

The questionnaire consisted of multiple sections, each of them containing a specific category of questions:

- General information, such as geographical location, job position, Agile role of the respondent.
- Practices, whose usage was measured through a Likert scale going from 1, "Used for every project", to 5, "Not used". Moreover, the respondents could also opt for 6, "Abandoned" or 7, "No experience, but I'd like to introduce it". These last two values were used in order to conduct a further analysis, but they still represent a subset of option 5.
- Variables, whose scale varied according to the specific item.
- Practices application, which allows identifying the determinants of practice changes over the respondents' experience.

Given that, in order to get an acceptable response rate (the time a respondent has to dedicate to the survey cannot generally be more than 8-10 minutes), it was necessary to make a selection of the practices with the purpose of reducing the answering time. A study made by Williams (2012) had already classified some practices according to their importance, given by the respondents of a survey. In this research, the 35 most important practices were selected and put in the questionnaire. This choice also gives room for further analysis: it would be possible to compare and analyse the differences between the importance given to a practice and its on-field utilisation.

The selection of Variables was based on previous literature, which tried to identify those conditions that might be responsible for the adoption and evolution over time of the practices. The Variables are: Team size, Team location and outsourcing, Organization size, Experience, Adoption rate, Decision making process, Industry.

The survey has been delivered by means of an online survey tool. Overall, 194 responses were collected and analysed.

A first, qualitative, statistical analysis allowed depicting the current state of Agile development, practices utilisation and inclination towards using Agile Methods. Then, a quantitative analysis took into consideration the correlation between the adoption of the practices and the success indicators. Both a factor analysis and a cluster analysis were performed with the software SPSS in order to: first, detect a common behaviour of the practices and reduce the analysis into a set of factors, which were associated to four explanatory concepts; secondly, divide the sample in clusters and perform a statistical analysis seeking for significant differences among the groups. Cluster analysis was based on the factor scores and allowed to compare the characteristic of the groups: statistically significant differences found in the variables would lead to identify a certain set of variables as determinants of a certain adoption strategy.

Results

Respondents were given the chance to pick multiple Agile methodologies, because the survey structure was based on the analysis of their overall experience with agility, and did not focus on the latest project they were involved in. Figure 1 shows the methodology prevalently used by the respondents. By far, Scrum is the most followed: considering the total answers, even three quarters of the sample indicated Scrum, which represents 32% of the total when the analysis is normalised to a basis of 100%. As can been seen in Figure 1, the Hybrid Scrum/XP is also largely applied, with a percentage equal to 31.8%.

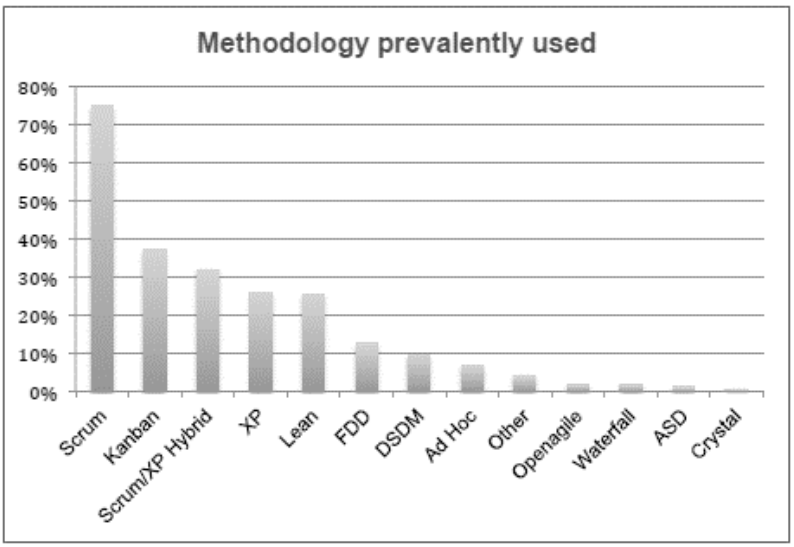

Fig. 1. **Usage of Agile Methodologies**
Source: Author construction based on survey

It is relevant to notice that the original Agile SDMs are not widely used anymore: methods such as, FDD, DSDM and ASD do not account for more than 7% overall. On the other hand, the application of new methodologies such as Lean and Kanban has increased its diffusion over the recent years; it is extremely remarkable that Kanban is the second most used

methodology after Scrum. Even if Kanban and Lean do not offer a complete guide to software development, the results are consistent with the question: as multiple answers were allowed, it is likely that many respondents combine the utilisation of these SDMs along with other, more structured and complete methods. For instance, a team could be XP-based and, at the same time, take inspiration from the Lean principles during some phases of the development.

A first conclusion can be drawn from the analysis of this question: most of the teams rely on methods that do not lead to relevant technical breakthroughs. The predominance of Scrum might imply that Agile adoption as affected mostly the managerial side of the process, whilst strictly coding related practices are still not so widespread (XP is only fourth). It seems that teams have largely embraced the agility principles without deeply transforming the practical activity of coding.

More than 65% of the sample is member of teams whose size is no bigger than 10 members. Indeed, one of the pillars of agility is communication among the team members, along with daily meetings and customer interaction. In order to effectively enhance communication, foster shared decision-making and shorten the length of the iteration, teams' members have to work close to each other.

Within the sample, about 65% is either co-located or works within the same building; this allows people to meet regularly and avoid any communication issue.

The core questions of the survey and of the analysis concerns the practices application. First of all, the application of the practices is extremely variable: not all the practices are used at the same time and most of the respondents seem to agree that only some of them deserve an intense application. Some practices are doubtless dominating.

Table 1

Usage of Agile Practices

	Used for every project	Used for more than 50% of projects	Used for about 50% of the projects	Used for less than 50% of projects	Not used	Abando-ned	I'd like to introduce it
Short iterations (30 days or	51,2	38,0	10,1	0,8	0	0,0	0,0
Prioritized product backlog	48,1	38,0	10,1	3,9	0	0,0	0,0
Stand up/Scrum meeting	46,5	43,4	6,2	3,1	0	0,0	0,0
Informal design; no big	39,5	35,7	17,8	6,2	0	0,0	0,0
Done criteria	34,1	19,4	31,0	9,3	2	1,6	2,3
Task Board	32,6	41,1	16,3	7,8	0	0,8	0,8
Iteration reviews/demos	31,0	26,4	20,9	16,3	3	2,3	0,0
Embracing changing	29,5	15,5	27,9	14,7	9	0,0	3,1
Complete features testing done during iteration	26,4	27,9	20,2	14,7	6,2	2,3	2,3
Requirements written as	24,8	42,6	29,5	3,1	0	0,0	0,0
Release planning	24,0	33,3	23,3	13,2	4	0,0	1,6
Burndown charts	21,7	26,4	26,4	21,7	0	0,8	2,3
Collective ownership of code	19,4	39,5	16,3	14,0	6	1,6	3,1

Features in iteration are customer visible/customer valued	14,0	28,7	26,4	17,1	12	0,8	0,8
Team velocity	14,0	25,6	28,7	21,7	6	0,8	3,1
Stabilization iterations	13,2	23,3	38,8	13,2	9	0,0	2,3
Design inspections	10,9	23,3	37,2	19,4	7	0,0	2,3
Emergent design	8,5	8,5	24,0	34,9	21	0,0	2,3
Frequent release of working software	7,0	11,6	24,0	31,0	21	2,3	2,3
Negotiated scope	7,0	7,0	17,8	34,9	31	0,8	0,8
Team documentation focuses on decisions rather than planning	7,0	16,3	31,0	24,0	14	0,8	7,0
Configuration management	6,2	20,2	35,7	24,0	11	0,0	2,3
Pair programming	3,9	3,9	20,2	37,2	20	0,8	5,4
Retrospectives	3,9	9,3	21,7	33,3	24	4,7	2,3
Automated unit testing	3,1	6,2	31,0	36,4	11	1,6	10,1
Sustainable pace	3,1	5,4	28,7	37,2	10	3,1	3,1
Whole multidisciplinary team with one goal	3,1	9,3	36,4	34,1	10	2,3	3,9
Planning poker	2,3	10,9	14,7	38,8	10	8,5	5,4
Test-driven development acceptance testing	2,3	3,9	11,6	29,5	28	3,1	20,9
Timeboxing	2,3	3,1	14,0	35,7	36	6,2	2,3
Synchronous communication	1,6	1,6	25,6	46,5	21	0,8	2,3
Test-driven development unit testing	1,6	4,7	12,4	37,2	24	3,1	17,1
Automated tests run with each build	0,8	7,0	30,2	33,3	14	0,8	13,2
Coding standard	0,8	0,8	13,2	32,6	51	1,6	0,0
Refactoring	0,8	0,0	7,8	37,2	42	2,3	9,3

Source: Author construction based on survey

By considering the standard deviation, high values are founded in Test-driven development, Embracing changing requirements, complete feature testing and Collective ownership of code. This gives evidence that their application is the most variable within the teams: some teams frequently use them, whilst others do not consider them worth the application.

Generally speaking, it is a bit surprising that Frequent release of working software, is not among the most used practices since it is even included in the list of the 12 principles of the Agile Manifesto.

The conclusion that can be drawn from this list is that the most used practices are generally related to soft development procedures. Instead of deeply modifying the way of working of each team member, they are mainly focused on improving the internal communication among the group members, planning and prioritising the development and,

overall, leaning towards informal design and simplification of the project management. On the other hand, taking into consideration the least used practices, it is quite surprising to notice that practices such as *Refactoring*, *Test-driven development unit testing* and *automated test run with each build* are located at the bottom of the list. They represent, indeed, some of the core practices of Agile coding processes and should –at least theoretically– be applied extensively.

In order to reduce data to a more manageable size while, at same time, retaining as much information as possible, factor analysis was applied to the 35 items that represent the various practices. In particular, the extraction methodology used was Principal Component Analysis (PCA). Prior to running the analysis, the suitability of data for this kind of analysis was assessed. As expected, the correlation matrix shown the presence of many items with a correlation higher than 0.3. Moreover, the Kaiser-Meyer-Olkin value of sampling adequacy was equal to 0.852. Also, Bartlett's test of sphericity reached statistical significance, supporting the factorability of the correlation matrix. The sample size and the absence of variables highly correlated to others (i.e. with correlation values above 0.9) allowed to keep all the 35 items in the analysis. PCA revealed the presence of eight components with eigenvalues exceeding 1, explaining cumulatively the 69.1% of the variance. A preliminary analysis of the component matrix, lead to think that an extraction with 8 components was not the most suitable solution for this case. Indeed, items loading strongly (more than 0.4) were detected within all the components, but their number was quite low for components 5, 6, 7 and 8.

An inspection of the scree plot revealed two possible breaks, respectively after the second and the fourth component. Therefore, it was decided to run a Parallel Analysis, which showed only four components with eigenvalues exceeding the corresponding criterion values for a randomly generated data matrix of the same size. Once made the decision to retain 4 factors only, factors were rotated with the Varimax method in order to improve their interpretability. Four common themes were identified.

Component 1. Process practices. Looking at the 11 practices that load strongly on component 1, almost all of them are related to the way of managing the development process. The main themes are the way people communicate, the length of the deadlines, the way of interacting with the customer and to prioritise the requirements. There are no practices that concern a more technical side of the development.

Component 2. Testing and coding practices. In this case, all the 8 practices are easily linkable to coding and testing techniques. These practices all attain to give practical instructions for writing the code and testing it with iterative methods.

Component 3. Principles practices. The theme of component 3 is quite subtle; even if some practices could initially be associated to component 1 as well, a further analysis allowed relating them to a broader and less practical theme. These 10 practices are, indeed, strongly connected to the 12 principles of the Agile Manifesto (AgileAlliance, 2001). They do not focus either on the process or on the coding techniques; they concern instead the general mind-set that should permeate the Agile users. For instance, Sustainable pace, Retrospective, Negotiated scope and frequent releases of working software are exactly included within the 12 principles. At the same time, they are practices as they imply certain behaviours of the practitioners.

Component 4. Team practices. Component 4 includes, for sure, 6 practices that are related to the team dynamics within the development. Actually, they have some shared points with the first three components, but the difference is that they are all related to teams, not individuals. This is proven by the presence of practices such as Team documentation focuses on decisions

rather than planning, Team velocity and Whole multidisciplinary team with goal. Even though these practices include process and coding characteristics, they all concern activities that can only be done by a team or that measure the overall outcome of the team. In the end, a punctual analysis of the loadings of each item led to some further modifications.

First of all, the minimum loading threshold was set to 0.4, meaning that items that loaded no more than 0.4 in every component were deleted and not considered in the analysis. Two practices, therefore, were erased: Planning poker and Team velocity. Secondly, two items had strong loadings on multiple components. It was the case of Synchronous communication, with a loading of 0.757 on Component 2 and of 0.421 on component 3. Having considered the managerial side of this practice, it was indeed quite ambiguous: this practice concerns both a technical point of view (as it implies the synchronicity of communicating the coding modifications and the possibility to see "live" what the other members are doing) and a principles point of view (as, also, implies that all the members meet in the same place to share the progress done). A similar evaluation was made also for Pair programming: this practice showed a loading equal to 0.538.

On Component 2 and equal to 0.546 on Component 3. In this case, there was a lack of consistency between the values and the actual meaning of the practice, since Pair programming is doubtless related to coding, but PCA included it within the principles, as the loading was slightly bigger. For these reasons, it was decided not to consider these two practices either.

In order to answer the second research question, it was decided to go further with the analysis and try to carry out a cluster analysis based on the final factor scores generated by PCA with 4 components.

In order to understand to what extent the selected variables could actually affect the practices adoption, cluster analysis was the best way to relate the factor scores extracted from PCA to the characteristics of the given sample. The procedure consisted of different steps: first of all, an ideal number of clusters was detected; then, clusters was measured and evaluated according to their practices utilisation; in the end, each variable was split and analysed cluster by cluster for the purpose of detecting statistically significant variables values. The suggested number of cluster obtained was 4. K-means was selected as the best method as it allows allocating each variable to a cluster of which the distance to the mean is the smallest. So, the analysis was carried out with K-means clustering method and manually setting the number of clusters to 4. All the factor scores had significance equal to 0, which implies that the model was valuable; considering the F values, factor score 1 is the one that contributed the most to the identification of the clusters. The number of cases per cluster ranged from 19% to 30%, thus indicating a good homogeneity. Anyway, in order not to be superficial, k-means cluster analysis was run also with other numbers of clusters. Considering that significance of the other cases was lower and, furthermore, no different conclusions could be drawn from those analyses, the decision was to keep 4 as the most suitable number of clusters.

Before going further with the analysis, each observation was linked to four different scores, each of them being the mean of the practices application of the practices included in one of the four different components individuated with PCA. This way, it was possible to get an overall assessment of the behaviour of the respondents for each component.

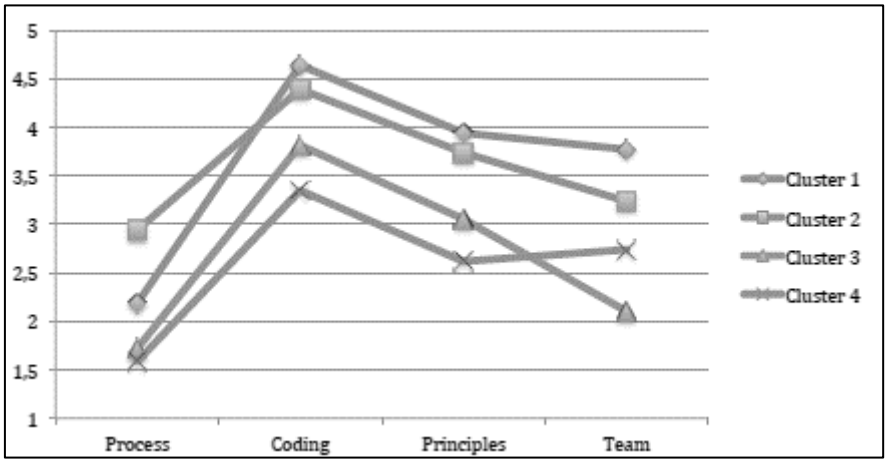

Fig. 2. **Application of practices by cluster.**
Source: Author construction based on survey

Apart for the scores of clusters 3 and 4 in the Team practices application and cluster 1 and 2 in Process, there is no significant overlapping; this means that even though practice application of the components is higher or lower among the groups, the application strategy is quite similar though. Moreover, it is extremely clear that practices related to the whole development process are by far the most used within all the respondents, while coding and testing practices are on average the least used. Anyway, there are some noteworthy differences. Group 4, for instance, seems to be made of respondents, which use the practices more intensively: it is always the one with lower scores, except for the team practices application, wherein cluster 3 is the leader. On the other hand, cluster 1 and cluster 2 are characterised by respondents who have a lower degree of practices application.

So, the last step of this analysis focused again on the elected variables. This time, though, each variable was split and separately analysed cluster by cluster. Before doing so, however, it was necessary to carry out a statistical analysis aimed to assess the significance of the differences of the means found for every cluster. Indeed, in case the Chi-Square tests or ANOVA analyses did not show significant results, the means would have to be considered homogenous for the sample analysed. Out of the 8 variables identified, only 2 of them turned out to have significant differences of the means among the clusters: Team size and Team location.

The first conclusion is that these two factors are the only ones that can actually influence the adoption of Agile practices. It is worthy, also, to consider the means differences shown in each cluster, and try to relate them to the scores of the practices adoption resulted from PCA. Clusters 1 and 2 are more likely to work in bigger teams, made of more than 10 members, whilst clusters 3 in and 4 report an average smaller size of the team. Moreover, means variation give a good explanation to the fact that, although cluster 4 was on average the best performer, cluster 3 had a higher degree of utilisation for the Team practices. It is no coincidence, indeed, that cluster 3 has the lowest mean: the lower the team members, the more team practices can be applied. Means for cluster 1 and 2 are quite similar, so was their practices application score. Results are consistent for these cases also: a big team size can tangle the extensive application

of practices. Speaking about team location, Cluster 4 is by far the ones characterised by a lower degree of de-location, while cluster 2 is the one that, averagely, has teams made of developers who work in different places. Also in this case, results are quite consistent with the initial assessments: agility requires frequent communication and interaction and it is facilitated by the proximity of the team members. Cluster 4, indeed, has a mean of 1.90 and its members are using practices more frequently than the others. The mean of cluster 3 is the second lowest and it is the cluster with the second best scores of Figure 2.

In the end, this study allows to answers to the second research question (What are the key variables that lead to a certain adoption strategy?): there are two key variables, team size and team location.

Finally, and consistently with the theoretical foundations of Agile, project members turned out to be the main subjects involved in the decision-making process. Precisely, three fourth of the survey base picked the project members in the multiple-answer question, proving that team, as a whole, is really the most important actor involved in the decisions about Agile development. However, the PM (or the analogue role for other methods) is said to be the decision-maker by 46.10% of the sample analysed.

Table 2

Decision making in Agile development

	Responses (N)	Percent	Percent of Cases
The project members	96	39,30 %	75,00 %
The coach/mentor/PM	59	24,20 %	46,10 %
Consultant	9	3,70 %	7,00 %
Company management	50	20,50 %	39,10 %
The responsible of the IT	26	10,70 %	20,30 %
Customer	4	1,60 %	3,10 %
Total	244	100,00 %	190,60 %

Source: Author construction based on survey

According to Agile suggestions, it means that not every team is applying correctly the Agile methods, either because they are still going through a transition phase or because they still believe that a unique decision-maker could be a better solution. Interesting and quite negative is also the high frequency of cases when the management of the company decides to adopt agile methods given that very often these decisions are based on word of mouth or just following the "new project management approach", than really understanding the implications.

Summary and conclusion

The Agile Manifesto formalised a series of values and principles, whose initial practical application could be found in methods such as Scrum, XP, FDD, ASD and DSDM. Empirical evidence has proven that, apart for Scrum, which turned out to be the dominant methodology, most of the initial models are not so widespread. The evolution of Agile is leading towards an intense adoption of practices related to the managerial side of the development, whilst more technical practices seem to be meant to a lower adoption.

The first research question was determined as follows: *What are the dominant Agile business practices?* First of all, there is set of 11 practices whose application is particularly frequently made by the respondents:

1. Stand up/Scrum meetings
2. Short iterations (30 days or less)
3. Prioritised product backlog
4. Informal design; no big design upfront
5. Task board
6. Requirements written as informal stories
7. Collective ownership of code
8. Iteration reviews/demos
9. Release planning
10. Complete feature testing done during iteration
11. Done criteria

More than half of the respondents declared to use these practices for more than 50% of the projects they are involved in. Moreover, some common characteristics were found: the majority of them come from the Scrum SDM, which indeed turned out to be the most used one; most of them, also, concern the managerial side of software development and do not give strict guidelines about the way of coding and testing the output.

Factor and cluster analysis allowed clarifying the second research question: What are the key variables that lead to a certain adoption strategy?

There are 4 different components to which practices can be associated:

1. Process related practices
2. Coding and testing related practices
3. Practices related directly to the principles
4. Team practices

The utilisation is quite different for the 4 groups: process related practices, once again, proved to be the most intensively applied. About the key variables, the only two that resulted to be statistically significant were Team size and the Degree of delocation: differences through these conditions influence the adoption strategy of the practices.

The third research question was: Why do the practices change within the PM groups?

Likely, the decision to deploy, modify or abandon practices is normally traced back to team members, and this is fully aligned with Agile principles. However, there is still a quite frequent influence of managers that define which practices should be used or dismissed.

Limitations and future research

This study is scoped only into respondents' opinions and their Agile experience. Limitations come from the nature itself of the research: a survey-based analysis was doubtless the best way to reach the goal of the study, nevertheless, the quality of the analysis could be affected by the sample size and the reliability of the answers. Collected data are, by definition, self-reported; hence poor memory or misunderstandings could contribute to inaccuracies. Also, even though sample size allowed a good quality factor and cluster analysis, a larger sample might have led to detect more significant key variables. Indeed, the biggest issue identified during the analysis was the lack of statistical significance means score differences of the variables among clusters.

Further research could take advantage of the analysis of the practices in order to relate their utilisation both with their theoretical importance and with their relation with the success of the development. Williams (2012) had already detected and classified these practices in terms of perceived importance; further research could look for a statistical correlation between theoretical importance and actual field usage of the practices. Moreover, this study did not focus on the possible correlation between a certain degree of practices utilisation and final success of the project. Finally, different key variables could be taken into consideration and analysed with the same procedure. Agile methods are still evolving and spreading over several different industries and countries: new potential variables might emerge in the future.

References

Abrahamsson, P., Conboy, K., & Wang, X. 2009. Lots Done, More To Do: The Current State of Agile Systems Development Research. *European Journal of Information Systems*, 18(4): 281-284.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. 2002. *Agile software development methods*. VTT Publications.

Ambler, S. W. 2002. *Agile Modeling: Best Practices for the Unified Process and Extreme Programming*. New York: John Wiley & Sons.

Anderson, D. 2010. *Kanban: Successful Evolutionary Change for your Technology Business.* Blue Hole Press.

Avison, D. E., & Fitzgerald, G. 2003. *Information systems development: methodologies, techniques and tools*. 3rd Edition. London: The McGraw-Hill Publishing Company.

Barlow, J. B., Giboney, J. S., Keith, M. J., Wilson, D. W., Schuetzler, R. M., Lowry, P. B., & Vance, A. 2011. Overview and Guidance on Agile Development in Large Organizations. *Communications of the Association for Information Systems*, 29 (July 2011): 25–44.

Batra, D. 2009. Modified agile practices for outsourced software projects. *Communications of the ACM*, 52(9): 143.

Beck, K., & Andres, C. 2004. *Extreme Programming Explained: Embrace Change*. 2nd Edition. Boston: Addison-Wesley.

Boehm, B., & Turner, R. 2005. Management Challenges to Implementing Agile Processes in Traditional Development Organizations. *IEEE Software*, 22(5): 30– 39.

Bose, I. 2008. Lessons Learned from Distributed Agile Software Projects: A Case Based Analysis. *Communications of the Association for Information Systems*, 23(December 2008): 619–632.

Conboy, K. 2009. Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20(3): 329–354.

Clutterbuck, P., Rowl, T., & Seamons, O. 2009. A case study of SME web application development effectiveness via Agile methods. *Electronic Journal of Information Systems Evaluation*, 12(1): 13–26. 96

Fitzgerald, B., Hartnett, G., & Conboy, K. 2006. Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems*, 15(2): 200–213.

Hayes, S. & Andrews, M. 2003. *An introduction to Agile Methods*.

Kong, S., Kendall, J. E., & Kendall, K. E. 2012. Project contexts and use of Agile Software Development Methodology in practice: a case study. *Journal of The Academy of Business & Economics*, 12(2): 1–16.

Kumar, A., & Goel, B. 2012. Factors Influencing Agile Practices : A Survey. International Journal of Engineering *Research and Applications*, 2(4): 1347– 1352.

Laanti, M., Salo, O., & Abrahamsson, P. 2011. Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology*, 53(3): 276–290.

Layman, L., Williams, L. & Cunningham, L. 2006. Motivations and measurements in an agile case study. *Journal of Systems Architecture*, 52(11): 654-667.

Lee, G., Delone, W., & Espinosa, J. A. 2006. Ambidextrous Coping Strategies in Globally Distributed Software Development Projects. *Communications of the ACM*, 49(10): 34–40.

Livermore, J. a. 2008. Factors that Significantly Impact the Implementation of an Agile Software Development Methodology. *Journal of Software*, 3(4): 31–36.

Mangalaraj, G., Mahapatra, R., & Nerur, S. 2009. Acceptance of software process innovations - the case of Extreme Programming. *European Journal of Information Systems*, 18(4): 344-354.

Nerur, S., Mahapatra, R., & Mangalaraj, G. 2005. Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5): 72–78.

Poppendieck, M. 2001. Lean Programming. *Software Development Magazine*, 9(5): 71-75.

Poppendieck, M., & Poppendieck, T. 2003. *Lean software development: An Agile Toolkit*. Boston: Addison-Wesley.

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

Salo, O. & Abrahamsson, P. 2007. An Iterative Improvement Approach for Agile Development: Implications from multiple case study. *Software Process: Improvement and Practice*, 12(1): 81-100.

Salo, O., & Abrahamsson, P. 2008. Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. *Software, IET*, 2(1): 58–64.

Schwaber, K. & Beedle, M. 2002. *Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice-Hall.

Schwaber, C., & Fichera, R. 2005. *Corporate IT leads the second wave of agile adoption*. Forrester.

Sidky, A., Arthur, J., & Bohner, S. 2007. A disciplined approach to adopting agile practices: the agile adoption framework. *Innovations in Systems and Software Engineering*, 3(3): 203–216.

Strode, D. 2006. Agile methods: a comparative analysis. In *Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualification* (pp. 257–264).

VersionOne. 2014. *State of Agile Survey*, Eight Annual Survey.

VersionOne. 2015. *State of Agile Survey*, Ninth Annual Survey.

Wang, X., Conboy, K., & Pikkarainen, M. 2012. Assimilation of agile practices in use. *Information Systems Journal*, 22(6): 435–455.

West, D., & Grant, T. 2010. *Agile development: Mainstream adoption has changed agility*. Forrester.

Williams, L. (2012). What agile teams think of agile Principles. *Communications of the ACM*, 55(4): 71–76.