

Progressive Outcomes: is it a handy approach to support agile methods process improvement?

Rafaela Mantovani Fontana^{1,2}, Sheila Reinehr¹, Andreia Malucelli¹

¹Pontifícia Universidade Católica do Paraná (PUCPR)
R. Imaculada Conceição, 1155, Prado Velho, 80215-901 Curitiba, PR, Brazil

²Universidade Federal do Paraná (UFPR)
R. Dr. Alcides Vieira Arcoverde, 1225, Jd. das Américas, 81520-260 Curitiba, PR,
Brazil

rafaela.fontana@ufpr.br, sheila.reinehr@pucpr.br, malu@ppgia.pucpr.br

Abstract. *While software engineering relies on reference models, such as CMMI-DEV, to improve its processes and gain maturity; agile teams that wish to sustain agility demand other guidelines to drive improvements. To address this need, in a previous study we have proposed the Progressive Outcomes Framework as a result of a research about how real agile teams evolve in agility. Although we could describe agile maturing mechanisms, we lacked a perception of how handy out approach would be to guide improvements. The objective of this study was, thus, to identify agile practitioners' perception about the framework. We performed a survey with agile practitioners in Brazilian conferences and analyzed collected data using descriptive statistics. Results show that agile practitioners feel comfortable in following non-prescriptive guidelines, but they still lack guidance on how to improve in agility.*

1. Introduction

Software process improvement is usually driven by reference models such as Capability Maturity Model Integration for Development – CMMI-DEV [CMMI Product Team, 2010]. According to CMMI-DEV, software development processes must be defined and quantitatively managed. When teams wish to use agile software development methods – which emphasize people and interactions over processes and tools [Beck et al., 2001] – agile practices must be adapted to suit CMMI-DEV requirements. These adaptations have thus become a topic of interest in research and practice [Silva et al., 2015], as benefits have been identified in the combination of both approaches [Sutherland et al., 2007].

Nevertheless, if agile teams wish to sustain agile principles and practices in the highest maturity levels, they usually cannot rely on these reference models. The detailed definition of processes hinders sustaining agility [Paulk, 2001; Lukasiewicz and Miler, 2012]. A number of agile maturity models have therefore been proposed in the last years [Lepänen, 2013; Ozcan-Top and Demirörs, 2013]. They usually focus on sustaining agility, but an issue still remains: they are highly based on CMMI-DEV structure, by prescribing practices and stages of evolvment [Fontana et al., 2015].

Prescribing practices and linear stages of evolvment is a way to guide improvement in agile software development. However, there is plenty of evidence in

literature that agile teams struggle to follow prescribed practices [Sidky et al., 2007; Kettunen, 2012; Schweigert et al., 2012; Fontana et al., 2014]. In a previous research we have, thus, proposed a novel approach for maturing in agile methods, which we called the Progressive Outcomes Framework, published in Fontana et al. (2015). Although we have followed a rigorous scientific process to propose this framework, we still lacked practitioners' perception to identify the utility of the framework in industrial context.

The objective of this study was, thus, to identify the perception of agile practitioners about the Progressive Outcomes Framework for agile software development processes improvement. We have conducted a survey with individuals from industry and identified how handy they believe our approach is. For research, this study contributes by presenting a method for a designed artifact evaluation and results that foster future studies. For practitioners, we provide insights on the applicability of our framework in industry.

This paper is organized as follows. Section 2 presents works related to maturity in agile software development. In Section 3 we describe the evaluated framework, as a summary of the results published in Fontana et al. (2015). Section 4 describes our research approach. Section 5 presents data analysis results and Section 6 and 7, respectively, discuss findings and conclude the paper.

2. Related work

Agile software development methods have been proposed mainly since the Agile Manifesto [Beck et al., 2001]. This manifesto stated a number of values and principles of these software development methods, in which: people and interaction are valued over processes and tools; working software is valued over comprehensive documentation; customer collaboration is valued over contract negotiation; and responding to change, over following a plan [Beck et al., 2001].

Examples of such methods are Scrum, Extreme Programming, and Feature-Driven Development, among others [Abrahamsson et al., 2003]. While these methods are highly focused on people interaction, the use of the established maturity models to guide processes improvement (as CMMI-DEV) might hinder sustaining agility in higher maturity levels [Paulk, 2001; Lukasiewicz and Miler, 2012]. Agile community has therefore proposed a number of agile maturity models, which focus on developing agile values instead of defining and controlling work processes.

Nawrocki et al. (2001), Lui and Chan (2005) and Benefield (2010) have proposed models to guide improvements with Extreme Programming, all of them based in levels of maturity. Other models suggest guidelines for improvement for agile in general, i.e. do not mention specific methods. These are the ones proposed by Packlick (2007), Sidky et al. (2007), Qumer and Henderson-Sellers (2008) and Patel and Ramachandran (2009). The one we found in literature that focuses exclusively on Scrum was proposed by Yin et al. (2011). We have also, in previous studies, proposed some stages for agile adoption [Fontana et al., 2014].

The study by Sidky et al. (2007) reports that, when their model was evaluated by practitioners, one of the critics was on the difficulty of prescribing practices for agile development. This evidence has also been shown in other studies [Kettunen, 2012; Schweigert et al., 2012; Fontana et al., 2014], confirming the need for guidance in agile

improvement in some ways that teams are left to adopt the practices and processes as they please.

Next section presents, then, our proposal for describing agile software development maturing, which addresses this need for new approaches.

3. The Progressive Outcomes Framework

In previous research, we have conducted four case studies to identify how agile teams mature and, thus, describe this maturing process. In these case studies, we analyzed how real agile teams evolved their practices. The data collected from four teams was consolidated in a framework we named as the *Progressive Outcome Framework* [Fontana et al., 2015].

Agile methods are recognized for being highly customized when used in organizational context [Bustard et al., 2013]. We have observed this fact in our research because the practices agile teams adopt in their maturing processes are indeed idiosyncratic. We could not identify such a pattern of practices or processes adoption, neither stages of maturity for agile software development improvement.

Instead, we identified a pattern in the outcomes teams pursue (highlighted in *italics* in this section). In their improvement process, agile teams pursue outcomes in practices, in the team, in deliveries, in requirements, in the product and in the relationship with the customer. Figure 1 shows the outcomes we identified for each of these categories. These outcomes may be achieved with whichever practices teams chose to adopt and not all outcomes are mandatory. Teams chase the outcomes that are relevant to their organizational context.

In practices category, we observed that agile teams start focusing on *Agile learning*, which means getting training, coaching and trying agile practices. Next, comes the *Sensemaking of work processes*. This is the moment when the team acts on the work processes to adapt it to its context, using their own experience and resources available. Next, comes the *Down to earth decisions* outcome. It is pursuit when teams start to use practices such as metrics collection to understand work processes and make more grounded decisions.

The team category was found to be the main category in the improvement of agile teams. This is the core category because this is where action is taken and decisions are made. Teams start as *responsive*, characterized by the need of close and directive leadership. These teams might evolve to what we called a *Confident team*, in which individuals start to have autonomy to make decisions. The next outcome, an *Assertive team*, is the one that decides when changes need to be made in the process and auto-organization emerges.

Deliveries also evolve in the maturing process of agile teams. Teams start with *Expected frequent finished coding*, when sprints are used to finish code, but this code is not ready for delivery. It evolves to *Expected frequent deliverables*, in which the code is tested and prepared for delivery, but not delivered yet. The deliveries in the end of the sprints are actually performed when the outcome *Expected frequent deliveries* is accomplished. This is when the team implements practices, uses tools and techniques that allows them to deliver in the end of the sprint, but they still delay sometimes. The

Defined frequent deliveries characterizes an accomplishment of being able to deliver through iterations and on time.

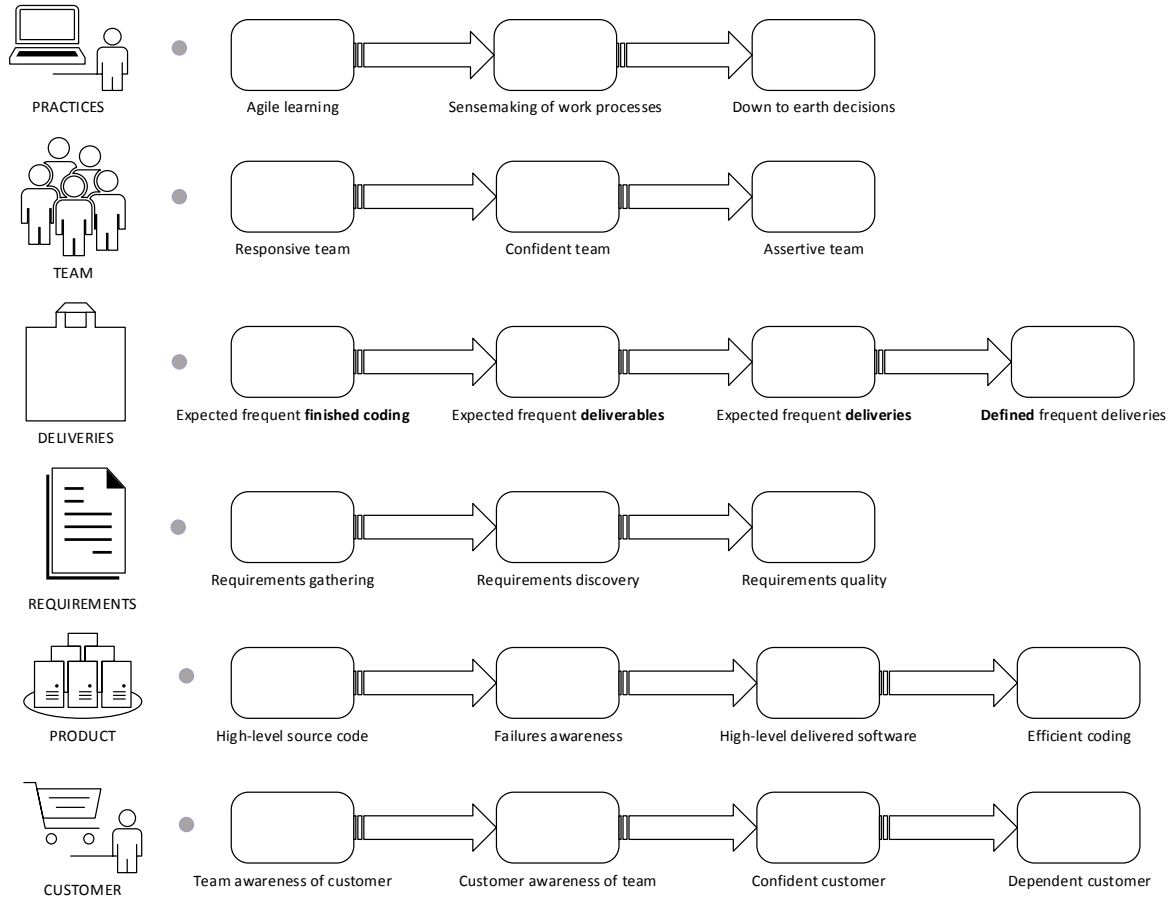


Figure 1. The Progressive Outcomes Framework. Published in Fontana et al. (2015)

The way requirements are elicited is also a target of improvement in the maturing process. The first outcome we identified was *Requirements gathering*, when, for example, big-up-front-requirements are collected and the team is not comfortable with changing requirements during the project. It evolves to *Requirements discovery*, when requirements are allowed to emerge. Another outcome agile teams pursue is *Requirements quality*, when practices to improve requirements quality are implemented, to raise the chances to meet customer needs.

In product category we placed the outcomes related to the software product itself. Teams start focusing in *High-level source code*, in which practices are implemented to guarantee the source code is robust. The *Failures awareness* outcomes represent the moment when team gets aware that even taking care of the code, bugs are delivered and new actions must be taken. Then, the focus is on *High-level delivered software*, in which endeavor initiatives to have the software tested and well prepared to be delivered with quality. The last outcome we identified was the *Efficient coding*. When pursuing this outcome, agile teams focus on engineering practices, such as test and integration automation, to make development activities more efficient.

Last but not least, the customer category describes the outcomes teams pursue in the relationship with the customer. We observed that the first two outcomes may be accomplished simultaneously – *Team awareness of customer* and *Customer awareness of team* – as it is the moment when team and customer get to know each other. With time, this customer gets more confident and respectful with the team, which we called the *Confident customer*. The next outcome is to have a customer that gets involved with the team in a level that customer might become a *Dependent customer*, which means that the team assists in customer’s business issues.

The dynamics of agile software development processes improvement has shown to be a process that cannot be described through maturity stages. It is a non-linear process of pursuit for outcomes and the practices used with this purpose vary between teams. In Fontana et al. (2015) we have shown that there is not even a sequence in the pursuit of the outcomes. Thus, it would not be surprising a team that used to adopt a plan-driven process to have accomplished the *Down to earth decisions* outcome before the others in the practices category.

These findings characterized how an agile team matures and the process improvement dynamics we identified. A gap remained, though, on how agile practitioners would perceive this framework. Next section describes, thus, how we structured this study to address this gap.

4. Research approach

The objective of this research was to identify the perception of practitioners about the Progressive Outcomes framework for software process improvement in agile software development. It is a designed artifact [Hevner et al., 2004] and, for this reason, we applied the suggestions given by Venable et al. (2012) for evaluation in design science research.

As a first step for a designed artifact evaluation, Venable et al. (*Ibid.*) states that the researcher should to analyze the context of the evaluation. With this respect, the evaluation of the framework is characterized by:

- Artifact: the guide for agile software maturity and the underlying maturity concept;
- Purpose: To “evaluate the formalized knowledge about the utility of a designed artifact for achieving its purpose” [Venable et al., 2012, p. 425];
- Ex-ante evaluation: it regards the evaluation of an uninstantiated artifact, given the infeasibility of testing the maturity framework in real settings within the research development period;
- Naturalist evaluation: we involved practitioners in the evaluation and, thus, explored the perceived performance of a situation within the organization;
- Method: the evaluation was performed using the survey approach [Forza, 2002].

The survey was applied in three agile software development conferences in Brazil: Agile Tour in Curitiba (September, 2014), Agile Tour in Campinas (October, 2014) and Agile Brazil in Florianópolis (November, 2014). In each of these events, we performed a fifty-minute speech, in which we explained the framework, each of its outcomes and, in the end, we emphasized the dynamics of agile processes evolvement,

as presented in Section 3. After the speech, we distributed printed questionnaires for the participants to evaluate the survey.

The questionnaire was based on Design Science Research guidelines [Hevner, 2004]. Hevner (*Ibid.*) suggests that when a designed artifact is produced in information technology research, it should be evaluated concerning its utility, efficacy and quality. As we had the premise that this evaluation should be quick and simple to be applied in a conference, we conceived two or three statements for respondents to evaluate each of these aspects, as shown in Table 1. Respondents evaluated each statement in a five-point Likert scale. The questionnaire also had a header to collect personal data: name, e-mail, company, city, position, experience time with software engineering and experience time with agile methods.

Based on the feedback we received from the two first events, and on the need to identify more qualitative perception of respondents, in the third event we added an open-ended question in the questionnaire. This question asked the respondent to comment or contribute to the framework presented.

Table 1. Statements evaluated in the questionnaire

Aspect to be evaluated	The Progressive Outcomes framework...
Utility	... is useful to aid teams to evolve with agile methods ... is useful to define what is maturing in agile
Quality	... is easy to understand ... comprises what I believe necessary to evolve in agile ... includes unnecessary information
Efficacy	... allows me to identify the current situation in my team/company ... is adaptable to different organization contexts

Collected data were analyzed using descriptive statistics. Answers given as “completely disagree” and “partially disagree” were grouped as “disagree”. The ones given as “completely agree” and “partially agree” were grouped as “agree”. Answers given as “no opinion” were kept the same way. We calculated, then, the percentage of respondents that agreed, disagreed and had no opinion about the statement.

As we are concerned about the applicability of the framework in practice, the experience of the respondent with agile methods was a key information. Thus, we analyzed responses considering all respondents and, separately, considering just experienced respondents responses, i.e., individuals with three or more years of experience with agile methods.

4.1. Threats to validity

The main threat to validity we identify in this study is the tendency of respondents to give positive feedback, considering that the majority of the statements in questionnaire pointed out to potential positive characteristics of the framework. To reduce this threat we added the open-ended question in the third event, allowing the respondent to comment positively or negatively, without bias.

5. Results

We obtained feedback about the Progressive Outcomes Framework from 231 agile practitioners, from which 93 had three or more years of experience with agile methods and, thus, were considered experienced practitioners. Respondents came from 31

different cities in Brazil. The experience profile of the respondents is presented in Table 2. It shows the percentage of respondents for each range of experience with software engineering in general and specifically with agile methods.

Most respondents had more than three years of experience with software engineering and, with agile methods, most respondents ranged from no experience to two years of experience. Considering that, in Brazil, most companies have less than five years of experience with agile methods [Melo et al., 2013], the percentage of 40% of respondents with more than three years of experience with agile methods is considered relevant.

Table 2. Respondents' profile

Experience with software engineering (years)	
0 to 3 years	25%
4 to 10 years	27%
> 10 years	37%
Not informed	10%
Experience with agile software development (years)	
0 to 2 years	57%
3 to 5 years	31%
> 5 years	9%
Not informed	3%

Next subsections present, for each statement evaluated, the perception of all practitioners and the perception of experienced practitioners. In the tables that present the data highest percentages are highlighted in bold.

5.1. The framework is useful to aid teams to evolve with agile methods

This statement identified whether respondents saw the framework as an aiding tool to evolve in agile methods practice. Table 3 shows that there is no relevant difference between the perception of all practitioners and experienced practitioners. Both present more than ninety percent of agreement on the utility of the framework.

Table 3. Respondents' perception about utility to aid evolvement

Respondents	Opinion	Percentage
All practitioners	Disagree	2.6%
	No opinion	3.9%
	Agree	93.5%
Experienced practitioners	Disagree	4.3%
	No opinion	3.2%
	Agree	92.5%

5.2. The framework is useful to define what is maturing in agile

Regarding the concept of maturity in agility, this statement presented similar percentages considering all practitioners' perception and just the experienced practitioners' perception. The great majority agree that the categories and outcomes the framework describes aid on defining what is maturing with agile methods, as shown in Table 4.

Table 4. Respondents' perception about definition of maturity

Respondents	Opinion	Percentage
All practitioners	Disagree	0.9%
	No opinion	3.5%

Experienced practitioners	Agree	95.2%
	Disagree	1.1%
	No opinion	3.2%
	Agree	95.7%

5.3. The framework is easy to understand

This statement evaluated one of the quality aspects of the framework. The perception of all practitioners and experienced practitioners were similar. Close to ninety percent agree that the framework is easy to understand, as presented in Table 5.

Table 5. Respondents' perception about easiness to understand

Respondents	Opinion	Percentage
All practitioners	Disagree	5.2%
	No opinion	6.5%
	Agree	87.9%
Experienced practitioners	Disagree	5.4%
	No opinion	5.4%
	Agree	89.2%

5.4. The framework comprises what I believe necessary to evolve in agile

Another evaluation of the quality of the framework is its completeness, i.e. whether it comprises the necessary concepts to evolve in agile. Considering this aspect, the perception of experienced practitioners revealed to be different. Table 6 shows that, while 83.5% of all respondents consider that the framework is complete, for the experienced practitioners, the number decreases to 79.6%. We believe this difference is due to the fact that more experienced practitioners have a wider background of practices and learnings. This experience might provide them ideas other than the ones presented in the framework, that could still aid evolving in agility.

Table 6. Respondents' perception about completeness

Respondents	Opinion	Percentage
All practitioners	Disagree	5.6%
	No opinion	10.8%
	Agree	83.5%
Experienced practitioners	Disagree	11.8%
	No opinion	8.6%
	Agree	79.6%

5.5. The framework includes unnecessary information

This statement evaluated whether the framework presented more information than necessary, another evaluation related to quality. The percentage of experienced practitioners which disagree was higher than the percentage considering all participants, as shown in Table 7. One aspect in this statement evaluation was that close to a quarter of respondents gave no opinion. We guess this is due to the fact that respondents felt insecure to assert that we presented more than what is necessary to evolve in agility, i. e. it would be risky to remove information from the framework.

Table 7. Respondents' perception about unnecessary information

Respondents	Opinion	Percentage
All practitioners	Disagree	55.8%
	No opinion	29.0%
	Agree	15.2%

Experienced practitioners	Disagree	58.1%
	No opinion	25.8%
	Agree	16.1%

5.6. The framework allows me to identify the current situation in my team/company

Regarding whether the framework is effective in supporting an assessment of the current situation of a team or a company, Table 8 shows that close to eighty percent of participants agree. Considering the responses of all practitioners, results in a percentage slightly higher. However, experienced practitioners have pointed out that the framework might lack a support for assessment, with 16.1% of disagreement with the statement.

Table 8. Respondents' perception about ability to assess the situation

Respondents	Opinion	Percentage
All practitioners	Disagree	8.2%
	No opinion	6.9%
	Agree	84.8%
Experienced practitioners	Disagree	16.1%
	No opinion	2.2%
	Agree	81.7%

5.7. The framework is adaptable to different organizational contexts

The adaptability to different organizational contexts was evaluated in this statement. Table 9 shows that percentages were similar when considering all responses and when considering just experienced practitioners ones. The majority of practitioners have pointed out to agree with the statement and a relevant number gave no opinion. Our guess is that people do not have enough background of working with agility in different contexts to support agreement in this statement. After all, agility adoption in Brazil is in early stages [Melo et al., 2013].

Table 9. Respondents' perception about adaptability to organizational contexts

Respondents	Opinion	Percentage
All practitioners	Disagree	5.2%
	No opinion	17.3%
	Agree	77.5%
Experienced practitioners	Disagree	8.6%
	No opinion	17.2%
	Agree	74.2%

5.8. Open-ended question data

In the third event where we presented the framework, we also collected the perception of practitioners using an open-ended question, as explained in Section 4. The consolidation of these responses and feedback received after our speeches showed that:

- Practitioners missed the evolution of the organizational context in the framework;
- More evidence was necessary to show how to accomplish each outcome; and
- They also lacked a clear description of management initiatives in the maturing process.

To sum up, results show that agile practitioners agree with the utility of a non-prescriptive framework to guide agile software development evolution. Regarding quality issues, the framework seems to be easy to understand, and include enough information about the maturing process. From effectiveness point of view, the perception is that the framework is flexible and allows an assessment of the current situation of a team.

In addition to that, qualitative suggestions, pointed out to the need to analyze the organizational context in the evolution process, to detail evidence on how to assess whether an outcome was accomplished, and to describe the management role in this improvement dynamics. Next section presents our discussions about the results.

6. Discussion

The objective of this study was to identify the perception of agile software development practitioners about the Progressive Outcomes Framework. This framework presents an approach to agile methods maturing processes not based on maturity stages nor practices adoption. Instead, the foundation is that agile methods processes improvement is based on a non-linear and dynamic pursuit of outcomes in practices, in the team, in deliveries, in requirements, in the product and in the relationship with the customer.

Our results have shown that agile practitioners felt comfortable with this approach, as the majority of respondents agreed that the framework is useful, effective and has quality of information. We believe that it confirms the fact the agile practitioners dislike following prescribed practices [Sidky et al., 2007; Kettunen, 2012; Schweigert et al., 2012; Fontana et al., 2014]. The approach of describing the outcomes real agile teams pursue has enabled agile teams to glimpse “where” they are in agile software development maturing process, without assessing practices, tools or processes.

Acknowledging that agile teams are complex adaptive systems [Vidgen and Wang, 2009], we recognized that the prescription of practices would hinder the agile characteristic of self-organization and emergency. Our framework instead defines a “semi-structure” to guide agile software development evolution, as suggested by studies of dynamic capabilities of organizations that need to adapt quickly to external demands [Eisenhardt and Martin, 2000].

In the open-ended responses we identified the need practitioners still felt to have more evidence on how to accomplish each outcome. It is an evidence that, although they dislike practices prescription, they feel the need to understand where they are and where to go in agile evolution. An agile processes improvement initiative in this sense should point it out, with the challenge of not stiffening work processes.

Our framework places people as the central role of agile maturing, as we have shown in Fontana et al. (2015). The compliance of agile practitioners with this approach shows that, in agile software development, processes are the outcome of what people do to get the job done [Adolph et al., 2012]. Agile processes improvement should therefore focus on agile people improvement, with a shared sense of purpose to reach specific outcomes. In this sense, we believe organizational ambidexterity studies point out ways to successfully combine adaptability (focus on people) with alignment (focus on specific outcomes) [Gibson and Birkinshaw, 2004; Fontana et al., 2015b].

Literature shows plenty of interest in maturing agile processes by applying reference models, such as CMMI-DEV [Silva et al., 2015] and the value of this combination is also verified [Sutherland et al., 2007]. This study has shown that, on the other hand, there are agile practitioners interested in guidelines for agile methods improvement that allow them to evolve using whichever practices they please, and enabling them to deliver value as their own business contexts require [Fontana et al., 2015].

7. Conclusions

This study has presented the perception of agile software development practitioners about the utility of a framework that describes how teams evolve with agile methods. As this framework presents a novel approach for processes improvement initiatives, this research contribution is showing how agile practitioners received this approach and the challenges we still face when developing agile processes improvement initiatives.

We identify to main limitations in this study. The first is that the evaluation of the framework was performed by practitioners who experienced or appreciate agile methods. The feedback from a community who has never experienced agile would be interesting to complement this evaluation. Another limitation of this study was that respondents were not argued about a certification of agile processes. Although we have identified they agree with the approach of not having prescribed processes, we lack the perception of whether a certification of this maturing status in agile would be of relevance to practice. Both limitations are thus, suggested topics for further studies.

This study is part of a doctoral thesis that has the objective to characterize maturity in agile software development.

References

- Abrahamsson, P., Warsta, J., Siponen, M., Ronjainen, J. (2003). “New directions on agile methods: a comparative analysis”. ICSE '03: Proceedings of the 25th International Conference on Software Engineering. 3-10 May. pp. 244-254. DOI 10.1109/ICSE.2003.1201204
- Adolph, S., Krutchen, P., Hall, W. (2012). “Reconciling perspectives: A grounded theory of how people manage the process of software development”. The Journal of Systems and Software, 85, 1269-1286. DOI: 10.1016/j.jss.2012.01.059.
- Beck, K. et al. (2001) “Agile Manifesto”. Available in <http://agilemanifesto.org/>. Accessed in 2015, May.
- Benfield, Robert. (2010) “Seven Dimensions of Agile Maturity in the Global Enterprise: A Case Study”. Proceedings of the 43rd Hawaii International Conference on System Sciences. pp. 1-7. Honolulu, HI. DOI 10.1109/HICSS.2010.337.
- Bustard, David; Wilkie, George; Greer, Des. (2013) “The Maturation of Agile Software Development Principles and Practice: Observations on Successive Industrial Studies in 2010 and 2012”. 20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems (EBCS). Apr 22-24. pp. 139-146. DOI 10.1109/ECBS.2013.11.
- CMMI Product Team. (2010) “CMMI for Development, Version 1.3 (CMU/SEI-2010-

- TR-033)". Software Engineering Institute, Carnegie Mellon University.
<http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>
- Eisenhardt, K. M.; Martin, J. A. (2000) "Dynamic Capabilities: What are they?"
Strategic Management Journal, 21, pp. 1105-1121.
- Fontana, Rafaela M.; Reinehr, Sheila; Malucelli, Andreia. (2014) "Maturing in Agile: What Is It About?" In: *Proceedings of the 15th International Conference, XP 2014, Rome, Italy, May 26-30*, pp. 94-109, DOI 10.1007/978-3-319-06862-6_7
- Fontana, Rafaela Mantovani; Meyer Jr., Victor; Reinehr, Sheila; Malucelli, Andreia. (2015) "Progressive outcomes: A framework for maturing in agile software development". *Journal of Systems and Software*. Vol. 102. pp. 88-108. DOI: 10.1016/j.jss.2014.12.032
- Fontana, Rafaela Mantovani; Meyer Jr., Victor; Reinehr, Sheila; Malucelli, Andreia. (2015) "Management ambidexterity: A clue for maturing in agile software development". In: Lassenius, C. et al. (Eds.): *XP 2015, LNBIP 212*, pp. 199-204. DOI 10.1007/978-3-319-18612-2_17
- Forza, Cipriano. (2002) "Survey research in operations management: a process-based perspective". *International Journal of Operations and Production Management*. Vol. 22. No. 2, pp 152-194. DOI 10.1108/01443570210414310.
- Gibson, C., Birkinshaw, J. (2004) "The antecedents, consequences, and mediating role of organizational ambidexterity". *Academy of Management Journal*, 47(2), pp. 209-226.
- Hevner, A. R.; March, S. T.; Park, J.; Ram, S. (2004) "Design Science in Information Systems Research". *MIS Quarterly*. Vol. 28. No. 1. pp. 75-105.
- Kettunen, Petri. (2012) "Systematizing Software Development Agility: Towards an Enterprise Capability Improvement Framework". *Journal of Enterprise Transformation*, 2(2). pp. 81-104. DOI 10.1080/19488289.2012.664610.
- Leppänen, Mauri. (2013) "A Comparative Analysis of Agile Maturity Models". In: R. Pooley et al. (eds.), *Information Systems Development: Reflections, Challenges and New Directions*. Springer Science+Business Media. New York. pp 329-343. DOI: 10.1007/978-1-4614-1951-5_27.
- Lui, K. M., Chan, K. C. C. (2005) "A Road Map for Implementing eXtreme Programming". In: M.Li, B. Boehm, and L. J. Osterweil (eds.): *SPW 2005, LNCS 3840*, pp. 474-481. DOI 10.1007/11608035_38.
- Lukasiewicz, K.; Miler, J. (2012) "Improving agility and discipline of software development with the Scrum and CMMI". *IET Software*, vol 6, iss. 5, pp. 416-422. DOI: 10.1049/iet-sen.2011.0193.
- Melo, C. O., Santos, V., Katayama, E., Corbucci, H., Prikladnicki, R., Goldman, A., Kon, F. (2013) "The evolution of agile software development in Brazil". *Journal of the Brazilian Computer Society*, vol. 19, issue 4, pp. 523-552, DOI 10.1007/s13173-013-0114-x.
- Nawrocki, J., Walter, B., Wojciechowski, A. (2001). "Toward Maturity Model for eXtreme Programming". *Euromicro Conference 2001, Proceedings*. 27th. 04-06 Sep. Warsaw. pp. 233-239. DOI 10.1109/EURMIC.2001.952459.

- Ozcan-Top, Ozden; Demirörs, Onur. (2013) “Assessment of Agile Maturity Models: A Multiple Case Study”. In: Software Process Improvement and Capability Determination, 13th International Conference, SPICE 2013, Bremen, Germany, June 4-6. Proceedings. pp 130-141. DOI: 10.1007/978-3-642-38833-0_12. 2013.
- Packlick, Jay. (2007) “The Agility Maturity Map – a Goal Oriented Approach to Agile Improvement”. Agile Conference 2007. 13-17 Aug. pp. 266-271. DOI 10.1109/AGILE.2007.55.
- Patel, C., Ramachandran, M. (2009) “Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices”. Int. J. of Software Engineering, vol. 2. no. 1. pp. 3-28. Available at <http://goo.gl/FGe0eE>
- Paulk, M., (2001). Extreme Programming from a CMM Perspective. IEEE Software. Vol 18. Issue 6. pp. 19-26. DOI 10.1109/52.965798.
- Qumer, A., Henderson-Sellers, B. (2008). “A framework to support the evaluation, adoption and improvement of agile methods in practice”. The Journal of Systems and Software. Vol. 81. Issue 11. pp. 1899-1919. DOI 10.1016/j.jss.2007.12.806.
- Schweigert, Tomas; Nevalainen, Risto; Vohwinkel, Detlef; Korsaa, Morten; Biro, Miklos. (2012). Agile Maturity Model: Oxymoron or the Next Level of Understanding. A. Mas. Et al. (Eds). : SPICE 2012, May 29-31, pp. 289-294. DOI 10.1007/978-3-642-30439-2_34.
- Sidky, Ahmed; Arthur, James; Bohner, Shawn. (2007) “A disciplined approach to adopting agile practices: the agile adoption framework”. Innovations in Systems and Software Engineering. Vol. 3. Issue 3. pp. 203-216. DOI 10.1007/s11334-007-0026-z.
- Silva, Fernando Selleri; Soares, Felipe Santana Furtado; Peres, Angela Lima; Azevedo, Ivanildo Monteiro de; Vasconcelos, Ana Paula L. F.; Kamei, Fernando Kenji; Meira, Silvio Romero de Lemos. (2015) “Using CMMI together with agile software development: A systematic review”. Information and Software Technology. vol. 58. pp. 20-43. DOI: 10.1016/j.infsof.2014.09.012
- Sutherland, Jeff; Jakobsen, Carsten Ruseng; Johnson, Kent. (2007) “Scrum and CMMI Level 5: The Magic Potion for Code Warriors”. Agile Conference 2007, 13-17 Aug. pp. 272-278. DOI 10.1109/AGILE.2007.52.
- Venable, J.; Pries-Heje, J.; Baskerville, R. (2012) “A Comprehensive Framework for Evaluation in Design Science Research”. In: Peffers, K.; Rothenberger, M; Kuechler, B. (Eds.): DESRIST 2012, LNCS 7286, pp. 423-438.
- Vidgen, R.; Wang, X. (2009) “Coevolving Systems and the Organization of Agile Software Development. Information Systems Research”. Vol. 20. n. 3, pp. 355–376. DOI 10.1287/isre.1090.0237
- Yin, A., Figueiredo, S., Silva, M. M. (2011) “Scrum Maturity Model: Validation for IT organizations’ roadmap to develop software centered on the client role”. ICSEA 2011, The Sixth International Conference on Software Engineering Advances. pp 20-29. 23-29 Oct. Barcelona. Available at <http://goo.gl/SklUZr>