An Agile Maturity Model for Software Development Organizations

Felipe Santana Furtado Soares
UFPE/CIn – Informatics Center – Federal University of Pernambuco
C.E.S.A.R - Recife Center of Advanced Studies and Systems
Recife, Brazil
furtado.fs@gmail.com

Silvio Romero de Lemos Meira
UFPE/CIn – Informatics Center – Federal University of Pernambuco
Recife, Brazil
srlm@cin.ufpe.br

Abstract—**The transition from traditional methods to agile methods and the changes needed to achieve real benefits from them are difficult to reach. The change affects not only the software development team, but also several areas of an organization and, first and foremost, requires a cultural change. In this context, this paper sets out to define a maturity model that will guide the setting up and running of agile methodologies, based on the Capability Maturity Model Integration (CMMI), in software development organizations. Given the research question considered, the method chosen is a systematic review of the literature, followed by a field study in software development companies. Thus, it is hoped that higher rates of success will be achieved when agile development values, principles and practices are adopted.**

Keywords-Agile metodologies; Maturity Model; Scrum; Lean; CMMI.

I. INTRODUCTION

In recent years, substantial transformations have been taking place in the software industry, driven by the demands of the market. Given this backdrop, there has been a demand for organizations to pay special attention to improving their software processes in the pursuit of greater competitiveness and productivity. Therefore, one of the challenges these organizations face is to acquire maturity in their development processes by setting up and running quality models that receive worldwide recognition [1].

At the same time, the market itself imposes deadlines that are more and more competitive and require great agility and high productivity from teams when using processes that bring these about and identifying activities that do not add value to the final product [2].

The challenge then becomes even more complex, as it includes meeting the requirements of a mature model, without spiking productivity, which is based heavily on the control variables of a software development project, while adopting practices of agile processes.

Capability Maturity Model Integration (CMMI) is an approach to improve processes that provides elements that are essential for an effective process. It brings together best practices that address development and maintenance activities, thus covering the entire lifecycle of a product from conception to delivery and maintenance [1].

In the late 90s, several agile methods emerged, including: Adaptive Software Development [34], Crystal [33], Dynamic Systems Development [35], Extreme Programming (XP) [36], Feature Driven Development (FDD) [37] and Scrum. All these methods use agile principles such as iterative cycles, rapid delivery of software that works and simplicity, as defined in the Manifesto for Agile Development [11].

Some authors advocate using the agile approach for managing projects that are conducted in complex environments characterized by many initial uncertainties, and in which there are difficulties in defining the scope and drawing up comprehensive plans, besides a high degree of changes and constant pressures to deliver results within short periods of time. However, the authors claim that the hindsights offered by the traditional project management methods should not be set aside but rather should be combined with the new practices put forward by agile methods [4].

However, some companies still have difficulties in implementing methodologies, either for lack of knowledge, or due to their difficulty in adapting these methodologies to the context of their projects [16]

In this context, after having obtained the correct definition of a maturity model, the expectation is that agile methodologies will be implemented in a systematic and organized way, with more likelihood of their being undertaken successfully. Thus, the main objective of this research is to define a maturity model so as to guide the setting up and running of agile methodologies, based on the CMMI maturity model, in software development organizations, thus resulting in higher success rates when agile development values, principles and practices are adopted.

The paper is divided as follows: Section 2 presents the background overview of CMMI; Section 3 focuses on describing the main agile methodologies and its benefits; Section 4 presents an initial discussion about a maturity model and agile methodologies, showing the difficulties in the transition to agile methods, a technical analysis and an a initial maturity model definition to guide the setting up and running of agile methodologies, based on the CMMI maturity model; The last section concludes this work in progress and presents the next steps.

II. MATURITY MODELS

According to Prado [5], maturity can be defined as "*a way to measure the stage that an organization is at in its ability to manage its projects.*"

The positive and expected results for the company, arising from its growth and maturity, will not come simply from the immediate application of techniques, tools and

ISBN: 978-1-61208-304-9

dissemination of concepts, nor should the best results be expected in the short term. All organizations undergo a maturation process, and this process has to precede excellence. The learning curve for maturity is measured in years [6].

CMMI lays down guidelines to improve the processes of an organization and its ability to manage the development, purchase and maintenance of products and services [3]. The model defines a path towards continuous improvement in terms of five levels of organizational maturity.

The CMMI-based improvement of processes has been accompanied by excellent quantitative results in costs, schedules, productivity, return on investment (ROI), customer satisfaction and product quality [7].

III. AGILE METHODOLOGIES

In the last ten years, agile methodologies have been gaining ground in the Information Technology and Communication market. Several studies have shown the good results achieved by these companies [12].

A. *The main agile methodologies*

Scrum is a framework for planning and monitoring a project that follows the principles of the Agile Manifesto. Since it is iterative and incremental, it works well in an environment of constant change. It supplies self-managing teams and proposes a form of flexible and adaptable work, not only in relation to the scope and requirements of a project, but also with regard to the exchange of teams, tools, programming languages, etc. [14].

XP is an agile methodology targeted on Software Engineering, and pays greater attention to programming than to management, as the former is the focus of Scrum, which is the reason why these methodologies are normally used together [15]. It was created by Kent Beck in 1996 and seeks to improve a software project by using five essential values: communication, simplicity, feedback, respect and courage.

Large numbers of tools and techniques have been developed to enable organizations to apply Lean concepts and ideas, many of which emerged from TPS (the Toyota Production System), for example, Kanban, JIT (Just in Time), Jidoka, Kaizen, etc. [23].

FDD is an agile methodology for management and software development that combines agile project management practices with a complete approach to object-oriented Software Engineering [24].

B. *Benefits of Agile Software Development*

Cohn [16] consolidated some surveys conducted in 2008 on the benefits of adopting agile software development related to the following matters: cost and productivity, employees' commitment and job satisfaction, time to market, product quality, and stakeholder satisfaction:

- A study conducted by Mah [25] of QSMA has been collecting metrics on productivity and quality for more than 15 years. He conducted a rigorous comparison between 26 agile development projects and a database of 7,500 development projects, mostly traditional ones. The agile projects studied ranged in size from 60 to 1,000 people;
- An extensive survey conducted by Rico [26] on agile projects summarizes 51 studies and academic research papers and gives the main percentage improvements in productivity, cost, quality, scheduling, customer satisfaction and return on investment;
- A survey conducted by the company Version One [13] with more than three thousand people. This is the largest ever survey on the state of adopting agile development. It is international in scope and is the most comprehensive overview of the use of agile development practices;
- A survey conducted by Scott Ambler in February 2008 with 642 people [12];

Regarding the comparison on productivity, research by Mah [25] reports that agile projects are 16% more productive with a confidence level which is statistically significant. These results were corroborated by the research studies below:

- Among the participants in the VersionOne survey [13], 73% found that being agile had improved processes (50%) or had significantly improved them (23%);
- Among the participants in the Ambler survey [12], 82% found that productivity was higher or much higher than before when agile methods were used and only 5% thought that productivity was lower or much lower.

In line with the above research, Rico [26] showed that the average increase in productivity was 88% and the average savings in development costs was 26%.

Regarding the time-to-market, agile teams tend to launch their products faster than traditional teams. VersionOne [13] reported that 64% of participants said that the time-to-market improved (41%) or significantly improved (23%). Mah [25] compared 26 agile projects to the QSMA database which has 7,500 projects and showed that their time-to-market is 37% faster.

IV. A MATURITY MODEL AND AGILITY

Methods, practices and agile techniques for software development promise to increase customer satisfaction [17] by producing higher quality software and accelerating development time [10]. Therefore, organizations that put great effort into improving their processes based on CMMI also believe that agile approaches can supply incremental improvements [20][18].

A. *Difficulties in the transition to agile methods*

The transition to agile methods and the changes necessary to obtain the benefits are difficult to attain. The change affects not only the software development team, but also several areas of the organization; for example, the commercial, marketing and financial areas [16].

Shore [27] and Fowler [28] point out that one of the failures when adopting agile methodologies is related to people: "it is the team that brings success or failure". He also points to the need to use some concepts. For example, for Scrum and XP to be applied together and not just one or the other, given that the former deals with management aspects, while the latter deals with engineering techniques of the product.

Anderson [8] points out that one of the difficulties when adopting these methodologies is associated with the way they are conducted by organizations.

B. *Technical Analysis*

When a technical analysis is made of models like CMMI and agile methodologies such as Scrum, for example, it is important to note that the perspectives they take are not the same. While maturity models feature a perspective of continuous improvement based on more abstract processes, and aim at meeting the objectives, agile methodologies are more focused on certain contexts and offer a greater level of detail on how to develop a software project.

Maturity models have a broad organizational vision, since they recommend a "path" for continuous improvement, defined in maturity levels. Each level involves various process areas, which include managerial and engineering matters. Conceptually, to be considered as adhering to one of these levels, an organization must meet the goals established for each process area. In addition to the objectives to be met, practices are recommended for each process area that, after having been well performed, immediately lead to goals being achieved.

To reach a CMMI maturity level, the organization must comply with all the process areas of the desired level. CMMI states that "*The only required component of the model is the statement of the specific or generic goal*". This makes it clear that the processes defined do not need to do exactly what is described in typical working products, subpractices and practices. The only requirement is achieving the goals of the process area [1].

C. *Agile Maturity Model*

Methods, practices and techniques for agile software development promise to increase customer satisfaction [18] by producing higher quality software and accelerating development time [10]. Therefore, organizations that have made a large effort to improve their processes based on CMMI, now also believe that agile approaches can supply incremental improvements [20][18].

Turner [29] comments that, despite the characteristics between agile methods and CMMI being distinct, both have specific plans for software development and pursue what is best so that the organization may produce quality software. Davis [30] reports that despite there being great controversy about the compatibility of Agile Development Methods (ADM) and CMMI, they are not mutually exclusive. He complements this by explaining that there is a place for ADM in CMMI and, more importantly, those who have adopted CMMI may consider adding ADM to their processes.

Paulk [19], lead author of the initial version of the SW-CMM, assessed XP in relation to 18 key process areas of the original SW-CMM. He concluded that XP partially or completely covers 10 of the 13 areas required to achieve Level 3, and is not an obstacle for the other three.

Boehm [17] presented the view that agile and disciplined processes exist on a continuum and can be combined as appropriate based on the risk factors specific to a project.

Jeff Sutherland, co-author of Scrum, reported on a highly productive project and claims that the combination of Scrum and CMMI is more powerful than each of them separately, and he includes guidelines on combining Scrum and CMMI [31].

According to Anderson [9], the way to achieve greater agility with CMMI is to realize that the practices are primarily consultative or indicative, and that to correspond to a CMMI evaluation, an organization must demonstrate that the goals of a process area are being achieved by evidence coming from practices.

In 2008, the Software Engineering Institute (SEI) published a technical report advocating the idea that agile development methods and CMMI best practices are not in disagreement with each other, and that the approaches can be combined successfully [32]. In 2010, the SEI published a book describing case studies that show the integration between CMMI and agile software development [21], but this book did not propose a new maturity model.

According to Marçal [22], it is possible to live peacefully with agile and maturity approaches. Challenges, however, exist and are focused on meeting principles contained in the two approaches. If on the one hand, practices of the maturity model may be added that are not considered in agile methodologies, the essence of these methods should not be unduly shaken. Of course, what the organization should keep in mind is the success of its projects in terms of time, cost and quality. To reach these goals, the flexible and conscious use of maturity models and agile methodologies is valid, provided this is based on an architecture of processes aligned to these goals and the organizational culture.

In this context, the main objective of this study is to define a maturity model so as to guide the setting up and running of agile methodologies, based on the CMMI maturity model, in software development organizations, which result in higher rates of success when agile development values, principles and practices are adopted.

This model is divided into five levels of maturity as follows:

- **Level 1:** initial stage where organizations do not use any methodology and their processes are unpredictable and reactive;
- **Level 2**: the stage where processes are characterized by project. There are processes for planning and monitoring a project, but the organization's vision is by project, i.e., there is no portfolio management of projects. At this level of maturity, setting up agile methodologies starts with Scrum (a focus on managing projects and prioritizing requirements) and a part of the methodology of FDD;
- **Level 3**: the stage where the processes are well defined and characterized by the Organization. There is a standard process with well-defined criteria to

instantiate them at every context of a new project. Engineering processes are implemented with the focus on XP, FDD and Kanban;

- **Level 4**: the stage where the processes are managed quantitatively with the focus on the agile metrics defined in Kanban and FDD;
- **Level 5**: the stage where the process is often optimized, with the focus on continuous improvement of the processes using the principles of Lean Software Development.

V. Conclusion and Future Studies

It was some years ago that agile methodologies became popular in organizations that were seeking environments that conduct software development in a faster and more flexible way. With the purpose of improving the results of software development projects, many organizations choose to introduce agile methods into their development processes. However, many do so in a disorganized way.

This work is part of a proposal for a doctoral thesis and its research methodology is divided into two stages. The first step uses the research instrument called an 'exploratory study', its main objectives being to validate (i) that agile methodologies and maturity models can be used together; (ii) and that there is a need for software development organizations to use a maturity model so as to implement agile methods. This is being undertaken by making a systematic review of the joint use of agile methodologies and maturity models together with a field survey with a view to validating this approach by means of interviews and questionnaires conducted with appropriate staff in some companies that are using agility with the CMMI maturity model.

The second phase will set out to validate the proposal for creating a maturity model for implementing agile methodologies. The main challenge of this validation is related to the possibility of applying the model in a software development company and defining what metrics can be collected before and after adopting the model. Furthermore, the challenge is about isolating the variables before and after measurement to assess whether, in fact, the use of the model contributed to the successful implementation of agile methodologies.

References

[1] CMMI-DEV, CMMI for Development, V1.3 model, CMU/SEI-2010-TR-033, Software Engineering Institute, 2010.

[2] B. Boehm, “A View of 20th and 21st Century Software Engineering”, In Proceedings of the 28th international conference on Software engineering (ICSE), ACM, New York, NY, USA, 12-29. DOI=10.1145/1134285.1134288, 2006.

[3] B. Chrissis, M. Konrad, and S. Shrum, “CMMI Guidelines for Process Integration and Product Improvement”, Second Edition, Addisson-Wesley, EUA, 2007.

[4] M. Dias, “Um novo enfoque para o gerenciamento de projetos de desenvolvimento de software”, São Paulo, 2005. Portuguese Version Only.

[5] D. Prado, “Gerenciamento de Projetos nas Organizações”, 2 ed. Belo Horizonte: Editora de Desenvolvimento Gerencial, 2003. Portuguese Version Only.

[6] G. Levin and H. Nutt, “Achieving excellence in business development: the business development capability maturity model”, [S.I]: ALLPM, 2005, White paper, Available at: http://www.allpm.com, 10/10/2013.

[7] D. Gibson, R. Dennis, and K. Kost, “Performance Results of CMMI-Based Process Improvement.” In: CMU/SEI-2006-TR-004, ESC-TR-2006-004, Pittsburgh, PA: SEI, CMU, 2006.

[8] D. Anderson, “Agile Management”: http://www.agilemanagement.net, 10/10/2013.

[9] D. Anderson, "Stretching Agile to fit CMMI Level 3", Agile 2005 Conference, Denver, July, 2005.

[10] D. Anderson, “Agile Management for Software Engineering - Applying the Theory of Constraints for Business Results”, Prentice Hall, 2003.

[11] K. Beck, *et al*., “Manifest for Agile Software Development”, http://agilemanifesto.org/, 2001, 10/10/2013.

[12] S. Ambler, “Agile adoption rate survey”. http://www.ambysoft.com/surveys/, 2008, 2010, 10/10/2013.

[13] VersionOne, "State of Agile Development Survey Results", http://www.versionone.com, 2008, 10/10/2013.

[14] K. Schwaber and M. Beedle, “Agile Software Development with Scrum”, New Jersey: Prentice Hall, 2001.

[15] H. Kniberg, "Scrum and XP from the Trenches". Estocolmo: C4Media Inc., 2007.

[16] M. Cohn, “Succeding with Agile: Software Development with Scrum”, Bookman, 2011.

[17] B. Boehm and R. Turner, “Management challenges to implementing agile processes in traditional development organizations”, IEEE Software, September/October, 2005, pp. 30-39.

[18] B. Boehm and R. Turner, “Balancing Agility and Discipline: A Guide for the Perplexed”, AddisonWesley, 2003.

[19] M. Paulk, “Extreme Programming from a CMM perspective”. IEEE Software, November, 2001, pp. 19-26.

[20] P. McMahon, "Extending Agile Methods: A Distributed Project and Organizational Improvement Perspective," CrossTalk, The Journal of Defense Software Engineering, vol. 18, issue 5, 2005, pp. 1619.

[21] P. McMahon, “Integrating CMMI and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement” (SEI Series in Software Engineering), 2010.

[22] A. Marçal, B. Freitas, F. Furtado, T. Maciel, and A. Belchior, “Estendendo o Scrum segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI”. CLEI, San José, 2007. Portuguese Version Only.

[23] M. Poppendieck. and T. Poppendieck, “Lean Software Development: An Agile Toolkit”, Addison-Wesley, Longman Publishing Co., Inc., Boston, MA, USA., 2003.

[24] J. Luca, “Solutions That Make A Difference”, http://www.nebulon.com/articles/index.html, 2008, 10/10/2013.

[25] M. Mah, “How agile projects measure up, and what this means to you”. Cutter Consortium Agile Product & Project Management Executive Report 9, 2008.

[26] D. Rico, “What is the ROI of agile vs. traditional methods? An analysis of extreme programming, test-driven development, pair programming and Scrum”, 2008.

[27] J. Shore and S. Warden, “The Art of Agile Development”. O'Reilly Media, Inc., Sebastopol, CA, EUA, 2008.

[28] M. Fowler, “FlaccidScrum”. http://www.martinfowler.com, 2009.

[29] R. Turner and A. Jain, "Agile Meets CMMI: Culture Clash or Common Cause," In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods XP/Agile Universe, 2002, pp. 153-165.

[30] C. Davis, M. Glover, J. Manzo, and D. Opperthause, “An Agile Approach to Achieving CMMI”, http://www.agiletek.com, March 2007, 10/10/2013.

[31] J. Sutherland, C. Ruseng, and K. Johnson., “Scrum and CMMI Level 5: The Magic Potion for Code Warriors”, The 12th annual European Systems and Software Engineering Process Group Conference European SEPG 2007 11-14th June, Amsterdam, 2007.

[32] H. Glazer, J. Dalton, D. Anderson, M. Konrad, and S. Shrum, “CMMI or Agile: Why Not Embrace Both”. CMU/SEI-2008-TN-003, November, 2008.

[33] A. Cockburn, “Crystal Clear: A Human-Powered Methodology for Small Teams”, 1st edition, Addison-Wesley Professional, 2005.

[34] J. A. Highsmith, "Adaptive Software Development: A Collaborative Approach to Managing Complex Systems", New York, NY, Dorset House Publishing, 2000.

[35] J. Stapleton, "DSDM: Dynamic Systems Development Method: The Method in Practice", Addison-Wesley, Boston, MA, 1997.

[36] K. Beck, "Extreme Programming Explained: Embrace Change", 2nd Edition, Boston, MA, Addison-Wesley Professional, 2005.

[37] S. R. Palmer and J. M. Felsing, "A Practical Guide to Feature-Driven Development", NJ, Prentice Hall, 2002.