AP3M-SW – An Agile Project Management Maturity Model for Software Organizations

Felipe Santana Furtado Soares
UFPE/CIn – Informatics Center – Federal University of PE
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 PE
Recife, Brazil
srlm@cin.ufpe.br

Abstract—**Applying agile methodologies in organizations whose processes are based on maturity models, such as Capability Maturity Model Integration (CMMI) or Organizational Project Management Maturity Model (OPM3) has been the focus of much controversy in the academic and in the software industry environment. The two approaches apparently have some fundamental principles and different bases, but on the other hand, adopting them jointly has increasingly become a reality for software organizations. However, the rush to reach maturity levels within deadlines that are shorter and shorter and the definition of heavy and inflexible processes, result in improvement projects with unique objectives of adherence to such models, often reflected in carrying out unnecessary activities and generating excessive documentation. In this context, agile methodologies are more appealing as they are lighter and this is inevitably related to their apparently offering a faster development at a lower cost of human effort. In this scenario, this paper puts forward a definition of an agile project management maturity model for software development organizations.**

Keywords—Project Management; Agile Methodologies; Maturity Model; APM; OPM3; CMMI.

I. INTRODUCTION

Currently, one of the challenges of software organizations is to acquire maturity in their development processes by means of implementing improvement projects based on recommendations of quality models recognized worldwide, such as Capability Maturity Model Integration (CMMI) [1].

At the same time, applying agile methodologies in organizations whose processes are based on maturity models, such as CMMI or the Organizational Project Management Maturity Model (OPM3) [2] has been the focus of polemical debate both in the academic world and in the environment of the software industry. The two approaches seem to introduce some fundamental principles and bases that diverge from each other but, on the other hand, adopting them jointly has become a reality for software organizations [3].

According to the *The Chaos Report* [4] between 2008 and 2010, the rate of projects categorized as 'Success' increased from 32% to 37%, while the rate of projects categorized as 'Cancelled' decreased from 24% to 21%. The rate of 'Challenged ' projects decreased from 44% to 42%. Among the reasons that the Report gives for this significant improvement found in 2010, in relation to 2008, the following can be highlighted:

- The use of agile processes has been growing. Currently, they represent 9% of all Information Technology projects and have been adopted in 29% of new applications under development. The Institute concludes that the growth in the rate of 'Success' is directly related to the increase in adopting agile methodologies;
- The reduction in the use of the processes that follow the 'Waterfall" lifecycle, known as traditional methods, has already accounted for nearly 50% of the number of new implementations. However, some companies are still having difficulties in implementing the methodologies, sometimes for lack of knowledge, sometimes because of the difficulty in adapting such methodologies to the context of their projects [6].

Nowadays, the competitive differential no longer lies in using such methodologies but rather in overcoming the challenges implementing them correctly and in the search for continuous improvement in software development processes [6]. Scrum, one of the methodologies that has gained most popularity, has been used in different ways, sometimes for lack of knowledge in its use, sometimes because it does not completely fit into the needs of companies. Other approaches are available, for example, the use of Kanban in software maintenance projects, in which features such as fixed iterations may not make sense for all projects [14].

Several studies report the adoption and growth of the use of agile methodologies in recent years. What can be perceived already is that organizations have consolidated their interest in them, the growth of the agile community, the high level of discussions, events, etc. [6].

Mike Cohn [6] states that seeking knowledge of agile methods has grown and that this cannot be considered a simple fad. However, what is observed is the difficulty that organizations have in implementing them, sometimes due to badly conducted adaptations, which strike at agile values and principles, sometimes due to the excessive restrictions that a methodology has and which cannot be fitted into the needs of certain projects.

According to Sidky *et al.* [15], it is observed that even with the growing number of companies that are seeking to adopt agile processes, there are still few studies that guide companies in this adoption. When organizations attempt to implement agile methodologies in a non-systematic way, projects end up having the same problems previously found in traditional methodologies.

ISBN: 978-1-61208-367-4

At the same time, the adoption of maturity models in project management has been growing in the world [2]. However, none of them is exhaustively focused on implementing an agile project management in software development organizations, even the CMMI is sufficient with all processes well defined because it doesn't address agile methods directly. Some of the most widely used models for example are: OPM3 (Organizational Project Management Maturity Model) [2], KPMMM (Project Management Maturity Model) [22], CMMI (Capability Maturity Model Integration) [1], PMMM Project Management Maturity Model) [24], MMGP (Maturity Model for Project Management) [23] and P2MM (PRINCE2 Maturity Model) [24].

However, if there is a clear motivation for using methodologies that promote agility in development, the search for certifications and adherence to maturity levels continues. Accordingly, strategies that result in maturity of processes based on agile principles have been a common target among software companies.

In this context, this paper aims to answer the following *question-problem*: with a view to increasing the success rate of software development projects, is a maturity model effective as part of the organizational strategy of implementing agile project management gradually and in a disciplined way?

To answer this question, this paper presents a maturity model that can guide software development organizations in implementing agile project management projects based on the existing main maturity models (CMMI and OPM3), while making use of the best practices of APM (Agile Project Management) [17] and Agile Methods (Scrum [7], FDD – Feature Driven Development [12], Lean [16] Kanban [14], Crystal [8] XP – Extreme Programming [11]), in a disciplined and gradual manner.

The paper is divided as follows: Section 2 presents the background overview of project management maturity model, agile methodologies and agile project management; Section 3 presents an initial discussion about an agile project management maturity model, showing the benefits of agile methodologies and the model components; The last section concludes this work in progress and presents the future studies, including the model validation research methodology.

II. Background Overview

A. Project Management Maturity Model

Over the years, organizations have been increasingly motivated to adopt quality models focused on the maturity of the software process. One of the reasons for this is associated with the fact that the improvement in the quality of software is widely associated with the adequacy and adherence of their processes to the high levels of this model [19].

Maturity may be defined as "*a form of measuring the stage of an organization's ability to manage its projects*" [23]. A maturity model, in accordance with OPM3 (2003) [2] is a conceptual framework, with consistent parts, which defines the maturity of an area of interest, for example, the organizational management of projects.

Figure 1 shows the timeline with reference to the main maturity models.

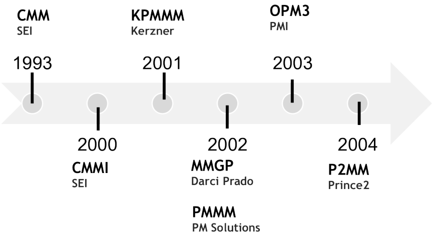

Fig. 1. Timeline with reference to the main maturity models.

The Capability Maturity Model (CMM) was originally developed by Watts Humphrey [20] and first appeared in his book 'Managing the Software Process'. He was inspired by the 20th century movement of manufacturing and quality assurance of the work of Juran, Deming and Crosby. The term "maturity model" and the five levels were inspired by Crosby's manufacturing model [28].

The CMMI is a process improvement approach that provides elements that are essential to an effective process. It brings together best practices that address development and maintenance activities, and covers the entire lifecycle of products from their conception to delivery and maintenance [1].

The KPMMM was created by Harold Kerzner and is set at five levels (Common Language, Common Processes, Unique Methodology, Benchmarking, and Continuous Improvement). It defines the current stage, planning and actions for implementing and gradually developing the management of projects [22].

The MMGP was created by Darci Prado and uses the same levels as the CMM Model. It possesses simplicity and universality (it is applicable to all types of organizations and to all categories of project) [23].

The PMMM of PM Solutions was created by Crawford also has five maturity levels and nine knowledge areas [24].

The OPM3 was established by the PMI (Project Management Institute). It is a model that aims to provide a path so that organizations understand their organizational management of projects and to measure the maturity based on a set of best practices in organizational project management. It describes a process in which the organization can develop or find a set of skills or good practices [2].

The P2MM was created by the Office of Government Commerce in 2006 and is based on the Project In Controlled Environments methodology [25].

B. Agile Methodologies

In the last decade, agile methodologies have been gaining space in the Information Technology and Communication market. Many studies show good results achieved by these companies, for example, research conducted by Scott Ambler reported a 55% success rate of projects, which used agile methodologies [26].

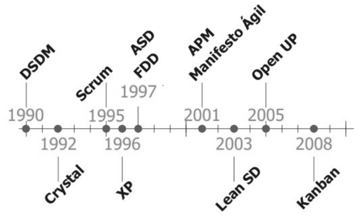

Fig. 2. Timeline referring to the main milestones which involve agile.

Figure 2 shows the timeline with reference to the milestones involving agile methods: Dynamic Systems Development Method (DSDM) [10], Crystal [8], Scrum [7], Extreme Programming (XP) [11], Adaptive Software Development (ASD) [9], Feature Driven Development (FDD) [12], Agile Project Management (APM) [17], the Agile Manifesto [27], Lean Software Development [16], Open Unified Process and Kanban [14].

Scrum is a framework established in 1996 by Schwaber and Sutherland and brings together monitoring and feedback activities, in general, in quick, daily meetings with the entire team, thus aiming to identify and correct any deficiencies and/or impediments to the development process [7]. Among the most used methodologies, Scrum appears as one of those that organizations most prefer (56%). In this same survey, the joint adoption with Kanban begins to be perceived [13].

XP was created by Kent Beck [11] in 1996 and seeks to enhance a software project using five essential values: communication, simplicity, feedback, respect and courage. Practices such as pair programming, rapid changes, constant feedback are core elements of the culture of this community.

FDD was created in 1997 in a large project in Java in Singapore. It arose from Coad's experience of object-oriented analysis and modeling, and Project management by De Luca. It is an agile methodology for managing and developing software, which combines the best practices of the agile management of projects with a complete approach to object-oriented Software Engineering [12].

Lean emerged in Toyota based on the idea that an increase in productivity is related to stopping doing anything that does not add value to the customer. Lean makes us think in a fast, uniform and quality flow without extra work that does not need to be done, without added defects [16]. Kanban brings the philosophy of Just in Time (JIT), which means producing only what is necessary, in the necessary time, in the necessary amount, and in the necessary location, and to do so with quality and the involvement of people, thus eliminating waste, and ensuring the continuous flow of production [14].

C. Agile Project Management (APM)

APM uses an empirical process model based on inspection and adaptation in order to promote exploration and an adaptive culture, to allow self-organization and self-discipline, to promote the reliability and consistency possible, given the degree of uncertainty and complexity inherent in the project, to be flexible and easily adaptable, allowing visibility throughout the process, to embed the learning, to encompass the specific practices of each stage and to provide points of verification [17].

According to Highsmith [17], *"[...] APM brings in itself a new focus on systems development, founded on agility, flexibility, communication skills and the ability to offer new products of value to the market, in short periods of time."*

The five stages of APM ***(Vision, Speculation, Adaptation, Exploration and Closure)*** were defined in order to promote the continuous delivery of value and to allow reflection that promotes learning. APM discards the anticipatory posture, based on prior planning actions and activities, characteristics of traditional project management, and seeks to develop a vision of the future and the ability to perform through situational exploration.

III. AP3M-SW – AN AGILE PROJECT MANAGEMENT MATURITY MODEL

According to Cohn [6], agile methodologies are generating significant gains in productivity with reductions in equivalent costs. This is due to several reasons: the adoption of these methods, which have mechanisms to release products on the market with much greater speed and to the satisfaction the client. In addition, they make it possible to visualize the development process better, which leads to greater predictability.

A. Benefits of Agile Software Development

Cohn [6] consolidated four surveys conducted in 2008 on the benefits of adopting the agile development of software related to the following aspects: productivity; time-to-market; and product quality: Mah [18] of QSMA, Rico [21], Version One [13] and Ambler [26].

Regarding the comparison on **productivity**, research by Mah [3] reports that agile projects are 16% more productive with a confidence level, which is statistically significant.

Regarding the **time-to-market**, agile teams tend to launch their products faster than traditional teams. VersionOne [13] reported that 64% of participants stated that the time-to-market improved (41%) or significantly improved (23%). Mah [18] compared 26 agile products to the QSMA database with 7,500 projects and showed that their time-to-market is 37% faster.

Regarding the **quality** of the product, Rico [21] states that agile teams develop higher quality products, based on 51 published studies on agile projects: a minimum improvement of 10% in quality and an average increase of 63%. According to Version One [13], 78% of participants responded that agile development improved (44%) or significantly improved (24%) software quality. In addition, 84% of the participants thought that the number of defects reduced by 10% or more.

However, according to Anderson [28], transition initiatives to agile methods may fail because prescriptive processes are powered by an organization to the delivery of the program as a part of the initiative and conducted by a process improvement group, an agile training group or a form of external consultancy. The workforce appears to tolerate the initiative, but actually passively resists this, because they believe that their unique situation does not fit into a standard process and the change is being forced, often without consultation or consensus.

In this scenario, the need therefore emerges to define a model that assists companies in implementing an agile management of projects in a more structured and mature manner.

B. Model Components

The way to achieve greater agility with the maturity model is to realize that the practices are primarily advisory or indicative, and that to correspond to an evaluation, an organization must demonstrate that the goals of a process area are being reached through evidence practice [28].

In this context, and based on the main models, frameworks and methods listed below, this study defines a maturity model for implementing the agile management of projects, the AP3M-SW, with the following features:

- It is based on **CMMI** to define five project management process areas with their respective specific goals and practices of software development, namely: **Project Planning** (to establish estimates, develop a project plan and obtain commitment to the plan); **Project Monitoring and Control** (to monitor the project against the plan and manage corrective action to closure); **Requirement Management** (to manage requirements); **Risk Management** (to prepare for risk management, identify and analyze risks and mitigate risks); and **Integrated Project Management** (to use the project's defined process and coordinate and collaborate with relevant stakeholders);
- It is based on the **OPM3** to define the domains of organizational project management **(Project, Program and Portfolio)** and the stages at which the organization is to be found **(Standardized, Measured, Control and Continuously Improved)**;
- It is based on the **APM** phases so as to define the project management process groups: **Vision, Speculation, Adaptation, Exploration and Closure**;
- It is based on **Agile Methods** (Crystal, Scrum, FDD, XP, Lean, and Kanban) so as to define practices and work products of each of the process areas.

Figure 3 illustrates the main components of AP3M-SW.

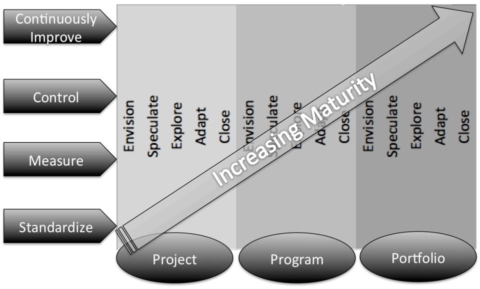

Fig. 3. Main components of the AP3M-SW Model (Adapted from [2]).

IV. CONCLUSION AND FUTURE STUDIES

The transition from traditional methods to agile methods and the changes necessary to obtain their real benefits are difficult to achieve. The change affects not only the software development team, but also various areas of the organization and, above all, this requires a cultural change.

Aligned to this context of the growth of the agile methods, the adoption of various maturity models in project management is growing worldwide. The challenge becomes how to combine these two approaches without losing their main features.

Various studies have already been conducted showing the possibility of getting on peacefully with agile and mature approaches [5]. If, on the one hand, it is possible to add practices of the maturity model not considered in agile methodologies, values and principles should not be compromised.

To guide companies who experience this scenario, this paper defined a maturity model that gives support in a disciplined and gradual manner when implementing agile project management based on relevant models, frameworks and already validated methods and on the community´s growing use of software development methods.

Future work is expected to detail all models' components and validate it. The main challenge of this validation is related to the possibility of applying the model in a software development company, through a case study methodology [29][30], and defining what metrics may be collected before and after adopting the model. Furthermore, isolating the variables before and after measurement to be able to assess if, indeed, the use of the model contributed to the success of implementing agile project management and, consequently, measuring the impact of this on the results of projects in relation to complying with the costs, time, scope, quality and satisfaction of the client and team. The time needed for implementation also presents a strong constraint, bearing in mind that maturity models need to be used gradually.

REFERENCES

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

[2] Organizational Project Management Maturity Model (OPM3®) Knowledge Foundation do Project Management Institute (PMI). 2003.

[3] F. S. F. Soares and S. R. L. Meira, "An Agile Maturity Model for Software Development Organizations", ICSEA, Venice, Italy, October, 2013, pp. 324-328.

[4] Standish Group. The Chaos Report. 2009, 2011. Available at: http://www.standishgroup.com. Last access: 08/2014.

[5] A. Marçal, B. Freitas, F. Furtado, T. Maciel, and A. Belchior, "Extending Scrum according to the CMMI Project Management Process Areas". CLEI, San José, 2007. Portuguese Version Only.

[6] M. Cohn, "Succeding with Agile: Software Development with Scrum", Bookman, 2011.

[7] K. Schwabber, "Agile project management with Scrum". EUA, Microsoft, 2004.

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

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

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

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

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

[13] VersionOne, "State of Agile Development Survey Results", Available at: http://www.versionone.com, 2008, 2013, Last access: 08/2014.

[14] D. J. Anderson, "Kanban – Succesful evolutionary change for your technology business". Blue Hole Press, Washington, 2010.

[15] A. Sidky, J, Arthur, and S, Bohner, "A disciplined approach to adoption agile practices: the agile adoption framework. Innovations In Systems

and Software Engineering, 2007, Volume 3, Number 3, pp. 203-216 (14).

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

[17] J. Highsmith, “Agile Project Management – Creating Innovative Products. EUA: AddisonWesley, 2004.

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

[19] D. Goldenson and D. Gibson, “Demonstrating the impact and benefits of CMMI: An update and preliminary results”, CMU/SEI-2003-SR-009. SEI. Pittsburgh, PA: SEI, 2003.

[20] W. S. Humphrey, “Managing the Software Process”. EUA: Addison-Wesley Professional, 1989.

[21] D. F. Rico, “What is the ROI of agile vs. traditional methods? An analysis of extreme programming, test-driven development, pair programming and Scrum”. TickIT International, 2008, 10(4), pp. 9-18.

[22] H. Kerzner, “Kerzner Project Management Maturity Model (KPMMM)”. Wiley, 2o edition. 2005.

[23] D. Prado, “Project Management Maturity Model (MMGP)”, Belo Horizonte: EDG, 2008. Portuguese Version Only.

[24] J. K. Crawford, “Project Management Maturity Model (PMMM)”, PM Solutions., 2o Edition, Center for Business Practices. 2007.

[25] PRINCE2 Maturity Model: Office of Government Commerce. 2006.

[26] S. Ambler, “Agile adoption rate survey”. Available at: http://www.ambysoft.com/surveys/, 2008, 2010. Last access: 08/2014.

[27] K. Beck et al. “Manifesto for agile software development”. Utah, 2001. Available at: http://agilemanifesto.org/. Last Access: 21/02/2012.

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

[29] B. A. Kitchenham and L. M. Pickard, “Evaluating software engineering methods and tools: part 9: quantitative case study methodology”. ACM SIGSOFT Software Engineering Notes, 1998, 23(1), pp. 24–26.

[30] R. K. Yin, “Case Study Research: Design and Methods”, EUA: Sage Publications, 2001, 200p.