

A Scrum-based approach to CMMI maturity level 2 in Web Development environments

C. J. Torrecilla Salinas
Consejería de Cultura. Junta de Andalucía
C/ Santa María la Blanca, 1
41004. Sevilla
carlos.torrecilla@iwt2.com

M.J. Escalona
Department of Computer Languages and Systems. University of Seville.
Av. Reina Mercedes s/n
41012. Sevilla
mjescalona@us.es

M. Mejías
Department of Computer Languages and Systems. University of Seville.
Av. Reina Mercedes s/n
41012. Sevilla
risoto@us.es

ABSTRACT

Scrum has become one of the most popular agile methodologies, either alone or combined with other agile practices. Besides, CMMI (Capability Maturity Model Integration) is accepted as a suitable model to measure the maturity of the organizations when developing or acquiring software. Although these two approaches are often considered antagonist, the use of an agile approach to reach certain CMMI maturity levels may result beneficial to organizations that develop Web systems, since they would take the advantages of both approaches. In Web community, this union may be very interesting, because agile approaches fits with the special needs of Web development, and they could be a useful tool for companies getting a certain grade of maturity. This work analyzes the goals of CMMI maturity level 2 and the feasibility of achieving them using the practices proposed by Scrum, trying to assess whether the use of this methodology is suitable for meeting the CMMI generic and specific goals or not. Finally, and based on this analysis, this paper raises a possible extension of Scrum, based on agile techniques, to accommodate the CMMI maturity level 2.

Keywords

CMMI (Capability Maturity Model Integration), Agile methodologies, Scrum, Web Engineering.

1. INTRODUCTION

Quick response, ability to adapt and change, short “time-to-market”, frequent user feedback... are some of the characteristics of Web development environment. In this kind of environments, agile software development methodologies are becoming a solid alternative for organizations developing software, as these methodologies offer a suitable framework for Web development characteristics [1]. The actual global and interconnected economy via Internet and the rise of the so-called Web 2.0 has brought the need of quick adaptation to the changing needs of markets and users, which is one of the strengths of agile methodologies [2]. Within the set of techniques, practices and methodologies that call themselves “agile”, Scrum is, by far, one of the most popular. It is worldwide used (either alone or combined with

other agile techniques) in approximately 70% of agile implementations [3].

By other hand, CMMI-DEV (Capability Maturity Model Integration for Development) is a well-established model in the software development field, which provides a comparative framework to measure the maturity level reached by an organization in this field [4]. It is mainly accepted that the accomplishment of the goals established by it is related with improvements in user satisfaction and product quality [5]. In general, agile approaches and maturity models proposed by CMMI have seen each other as opposite. However, both approaches include valid principles for software development that are not necessarily incompatible [2]. Recently, different initiatives have emerged trying to use agile practices to help organizations reach certain CMMI maturity levels, both from agile and CMMI sides. Our paper tries to help clarifying this issue. As Scrum is one of the most popular agile methods [3], we will use the set of practices proposed by it to study the compatibility with the mentioned maturity level 2. Based on the foregoing, our work aims to cover the following objectives: (1) To map the Scrum practices with the goals of CMMI-DEV level 2, in order to assess the viability of a Scrum approach to reach this maturity level. (2) To propose an extension of Scrum with a set of agile techniques with the aim of helping organizations reach CMMI-DEV maturity level 2.

To reach the above-mentioned objectives, an analysis of the related works was conducted and completed with our own study, with the aim of testing the possibility of reaching CMMI maturity level 2 with Scrum practices. The study consisted of a detailed analysis and study of Scrum guide [7] and CMMI-DEV version 1.3 model [4]. After the study, a structured search was conducted, identifying and analyzing the published literature related with the proposed problem, in order to identify the actual “state-of-art” and gaps. Finally, we analyzed every generic and specific goal for CMMI maturity level 2 and the feasibility of achieving them using the practices proposed by Scrum, and established a comparison between our conclusions and the published literature, in order to validate them. This paper is organized into the following sections. Following this introduction, Section 2 offers an overview of Scrum and CMMI-DEV level 2. Section 3 presents the problem addressed and describes the considered approaches. Section 4 includes a detailed study of the problem. Section 5 proposes an extension to Scrum to fit CMMI-DEV level 2. Finally, Section 5 states a set of conclusions and contributions and proposes possible future work.

2. AN OVERVIEW OF SCRUM AND CMMI-DEV

2.1 Scrum

Originally developed by Jeff Sutherland and Ken Schwaber [15], Scrum is an incremental and iterative framework for projects and it is one of the most popular agile methodologies. It divides the

development process into working cycles called *Sprints*, lasting from 2 to 4 weeks. These Sprints are time-boxed and must be carried out respectively without interruptions, although the planned work has not been completed. At the beginning of each Sprint, a multi-disciplinary team selects what requirements will be implemented from a prioritized list called Product Backlog. This team is in charge of fully developing those requirements by the end of the Sprint. During the Sprint, the selected items cannot change. Every day, the team members check their progress and adjust the next necessary steps to completely finish the work. At the end of the Sprint, the team together with the stakeholders reviews the results. This demonstration provides the feedback that enables the team to create new features to be included in following Sprints. Periodically, the team reflects on the development process to figure out how to improve it (through a retrospective meeting or a similar tool).

2.2 CMMI-DEV level 2

CMMI can be defined as a process improvement approach that provides organizations with the main tools to develop effective processes [2]. There are several CMMI models among which CMMI-DEV (whose latest version is 1.3) is the model for software development. It provides a set of best practices for services and products development, covering their complete life cycle from conception to deployment and maintenance. CMMI-DEV is composed of 22 process areas, out of which 16 are common to all CMMI models, 1 is shared and 5 are specific for software development. CMMI-DEV uses the levels to recommend the best way for an organization to evolve the process used to develop products and services. Levels are obtained as the final result of a formal evaluation process. One of the two improvement paths proposed by CMMI-DEV is the staged representation, which focuses on the maturity level of an organization, that is, a way of characterizing its performance. Each maturity level allows improving an important subset of processes into an organization, preparing it to the next level of maturity. Maturity levels are reached by achieving generic and specific goals related to a predefined subset of process areas. The CMMI-DEV maturity level 2 is called Managed. In this level, processes are planned and executed according to a policy by employing qualified personnel with suitable resources to produce controlled outputs. These processes are monitored, reviewed and controlled.

3. CONSIDERED APPROACHES

Successful software processes development will be given by the development team's capabilities of adequately manage complexity, technological innovation and requirement changes. Both agile methods and CMMI face these challenges from a different point of view and with different methods [16]. A systematic search [17] in published literature was conducted based on the following search strings: "agile, cmmi, maturity, model, agility", "agile, cmmi, maturity, model", "scrum, cmmi, maturity, model", "scrum, cmmi". It shows that, since 2005, several works have studied the compatibility between CMMI and agile methodologies, mainly Scrum and XP [2] [8] [9] [10] [11] [12] [13] [16]. Moreover, although we have found partial studies on the feasibility of an Scrum-based approach to CMMI maturity levels [8] [11] [12], we haven't found any study that provides an agile framework for the attainment of all generic and specific goals proposed by the level 2 CMMI maturity in Web Environments. Our search also shows that it can also be found studies about how agile organizations go toward CMMI accreditation [18] [19], and about organizations that have started appraisal process in certain CMMI maturity levels based only on agile practices [20]. Finally, it also can be found guidelines to gain agility for organizations with a certain CMMI maturity level [21].

Based on the reviewed literature, it can be extracted the following assumptions, confirming the possibility of an agile approach to CMMI maturity levels:

- *CMMI is a model*, neither a standard nor a methodology, establishing goals to achieve, but not how to reach them.
- *CMMI defines process areas not processes*, with the only requirement of satisfying the goals of each area [9].
- *Agile techniques have no lack of discipline or control*, on the contrary, delivering results in short periods of time, requires great discipline.
- *Agile techniques can be escalated*, as there are works that show examples of successful projects with more than 250 people [22].

As mentioned above, it is widely assumed that satisfaction of the highest CMMI levels is directly related to improvements of the quality of the software developed and user's satisfaction [5]. On the contrary, the achievement of CMMI highest levels can be very expensive for medium and small organizations, which are very common in Web environments. Furthermore, nowadays organizations developing Web systems are required to quickly adapt themselves to technological and business changes, which would be difficult to achieve with strict and rigid project management approaches [12]. An agile-based approach to CMMI would allow an organization, despite its size, first, to use the CMMI model to demonstrate a certain maturity level to customers and users based on a widespread model, and second, to maintain a quick response and adaptation capability, which is essential in an environment as changing as Web environment. Based on the foregoing, it can be affirmed that agile-based approaches to CMMI are not only possible, but also necessary and desirable [12], since organizations will benefit from both models. The main objective would be that CMMI should establish "what to do" and agile "how to do it". As mentioned in section 1, due to the fact that Scrum is one of the most popular agile methodologies [3], our study will be addressed to its proposed practices, and we will start by the first level of CMMI staged representation, which is level 2.

4. RESULTS OF THE STUDY

In regard with the approaches presented in Section 3, a relation between the proposed practices of Scrum and the goals of CMMI-DEV level 2 process areas has been established. It aims at determining whether a CMMI-DEV goal can be reached using only the practices proposed by Scrum. After this analysis, our conclusions were checked against the results of similar studies [8] [9] [10] [11] [12] [13]. Our results are presented in the following tables, containing every generic and specific goal for each process area, and represented with the following symbols:

- + The goal is completely covered by Scrum practices.
- * The goal is partially covered by Scrum practices.
- The goal is not covered by Scrum practices.

Table 1. Results for CMMI level 2 generic goals.

Goal	CM	MA	PMC	PP	PPQA	REQM	SAM
GG1 Achieve Specific Goals	-	+	+	+	-	+	-
GG2 Institutionalize a Managed Process	-	*	*	*	*	*	-
GG3 Institutionalize a Defined Process	-	*	*	*	*	*	-

In relation to the results of the analysis, it is shown that the exclusive use of Scrum cannot cover all the generic and specific goals of CMMI-DEV level 2, although it covers most of the specific practices of PMC,

PP and REQM process areas. According to the study, MA and PPQA process areas are only partially covered by Scrum, due to the fact that it does not define a systematic measurement plan and does not include specific practices related to product quality assurance. Additionally, CM and SAM process areas are not covered by any Scrum proposed practices. It is thought that both models may be compatible, after looking at these results and observing that Scrum can reach, either completely or partially, more than half of the generic and specific goals of CMMI-DEV maturity level 2.

Table 2. Results for CMMI level 2 specific goals.

P. Area	Goal	Result
CM	SG1 Establish Baselines	-
	SG2 Track and Control Changes	-
	SG3 Establish Integrity	-
MA	SG1 Align Measurement and Analysis Activities	*
	SG2 Provide Measurement Results	*
PMC	SG1 Monitor the Project Against the Plan	*
	SG2 Manage Corrective Action to Closure	+
PP	SG1 Establish Estimates	+
	SG2 Develop a Project Plan	*
	SG3 Obtain Commitment to the Plan	+
PPQA	SG1 Objectively Evaluate Proc. and Work Products	*
	SG2 Provide Objective Insight	-
REQM	SG1 Manage Requirements	+
SAM	SG1 Establish Supplier Agreements	-
	SG2 Satisfy Supplier Agreements	-

5. EXTENDING SCRUM

If, as it is shown in Section 4, Scrum is compatible with CMMI-DEV, it can be extended somehow to support the generic and specific objectives that are not completely covered by it. If that extension is made by means of other agile methods, it will be possible to maintain the flexibility and quick response capability that they provide to Web development projects and organizations. The objective of this extension will be, therefore, to extend Scrum without losing agility to completely satisfy the generic and specific objectives of CMMI-DEV level 2. As Scrum is one of the most accepted and popular agile framework, its extension to reach the CMMI-DEV goals can take several advantages [13], some of them are listed below:

- *A better risk management*, which could remove impediments and increase teams' productivity.
- *A systematic product quality management*, which will enable obtaining better products at the end of each Sprint.
- *A quicker and easier institutionalization of agile practices* within organizations.

The proposed extension will combine practices and techniques from other agile frameworks (like XP) and also ad-hoc modifications from the Scrum framework. The following table shows the proposed extension for Scrum to reach all the generic practices of CMMI-DEV maturity level 2:

Table 3. Proposed Scrum extension to cover generic goals of CMMI maturity level 2.

Goal	Proposed extension
GG1	The aforementioned extensions to reach the specific goals will guarantee the satisfaction of this generic goal.
GG2	In addition to the proposed extensions and to guarantee the involvement of the higher level of management, a project status report can be generated at the end of each iteration, including elements like finished user histories, team velocity, sprint burn down

	chart, relevant issues or results of established measurements, among others. This report could be generated through collaborative tools (as Wiki pages) or paper and posted as an "information radiator". The higher level of management will be also permanently invited to the Sprint review meetings, as relevant stakeholders.
GG3	To guarantee the institutionalization of a defined process, all aspects will be documented in a collaborative way, which assures the team-ownership of the process. All of the conclusions of the retrospective will be stored in a lessons learned backlog, which will be periodically revised.

The following table shows the proposed extension for Scrum to reach all the specific practices of CMMI-DEV maturity level 2. Extensions are proposed to a process area if any of its specific goals is partially or no covered by Scrum practices:

Table 4. Proposed Scrum extension to cover specific goals of CMMI maturity level 2 in each process area.

P. Area	Goal	Proposed extension
CM	SG1	eXtreme Programming: Shared code ownership, continuous integration and version control.
	SG2	
	SG3	
MA	SG1	Ad-hoc extension: Establish measurement objectives, how to measure them and how to store the measures during Sprint 0. Collect the measures during the Scrum meetings every Sprint. Review the measures during Sprint Retrospective. Raise an impediment through the Scrum Master when a problem is identified.
	SG2	
PMC	SG1	Ad-hoc extension: Establish how, when and where to store the project data during Sprint 0. Collect the data during the Scrum meetings every Sprint. Review the accomplishment of this agreement during the Scrum meetings. Use collaborative tools (like Wiki pages) to manage the data.
	SG2	
PP	SG1	Ad-hoc extension: Establish how to communicate and manage the project data during Sprint 0. Collect data during the Scrum meetings every Sprint. Review the accomplishment of this agreement during Scrum meetings. Use collaborative tools (like Wiki pages) to manage the data.
	SG2	
	SG3	
PPQA	SG1	Ad-hoc extension: Establish quality objectives during Sprint 0, briefly documenting the agreements. Include quality related information in product backlog to facilitate its traceability. Collect data related to quality from Scrum meetings, and raise, through the Scrum Master, any detected issue. Use collaborative tools (like Wiki pages) to manage the data.
	SG2	
REQM	SG1	No extension proposed , fully covered by Scrum
SAM	SG1	Agile contracts techniques: Use of agile contracts techniques to elaborate any Request For Proposals and select the contractor
	SG2	

As stated before, the proposed model combines Scrum with practices and techniques of other agile frameworks (XP and agile contracts) and includes some ad-hoc modifications to achieve all the specific and generic goals of CMMI-DEV maturity level 2. These modifications are based on the existence of a so-called "Sprint 0", a time-boxed iteration that occurs at the beginning of the project; where the team establishes the general basis for quality assurance, project data management and project measurement and analysis. This iteration is also used to generate the needed Requests for Proposals in an agile way. In Sprint 0, the general basis and rules are fixed: for example, how to measure quality assurance or how to collect, store and manage project data. If it is decided to outsource any part of the project, the corresponding Request for Proposals will also start in this iteration. Once this time-boxed phase has ended, the normal Sprint-guided life

cycle of Scrum will start, using, in each Sprint, in addition to the standard Scrum practices, all the proposed extensions to the model. Scrum meetings will cope with collecting data and higher-level information on the project status. At the end of each Sprint, some reports will be produced (by means of paper or collaborative tools, or posted as an “information radiator” into the team’s room) to keep the senior managers informed and involved in the process.

6. CONCLUSIONS AND FURTHER WORK

In this work, the viability of a Scrum-based approach to CMMI-DEV maturity level 2 for Web development environments is analyzed and the benefits of the combined use of both approaches are presented. As a starting point, a mapping between Scrum practices and CMMI-DEV level 2 specific and generic goals was performed, basing on the analysis of existing literature and our own work. It is shown that an approach to CMMI level 2 exclusively based in Scrum is insufficient to reach all the goals of this level. It is possible to point out that the use of Scrum could reach more than a half of CMMI-DEV level 2 specific and generic goals, either globally or partially. This fact leads us to conclude that Scrum could be compatible with CMMI-DEV model and it could serve as a starting point to reach the stated maturity level. In accordance with this conclusion and with the aim of maintaining an agile set of practices, a Scrum extension that combines Scrum with practices of other agile frameworks and includes some ad-hoc modifications was proposed. This extension adds to Scrum tasks that could be included mainly in Sprint 0. The systematic execution of the proposed tasks, together with Scrum practices, could allow an organization to show evidences of achieving the CMMI-DEV level 2 goals. The presented proposal can, in our opinion, help an agile organization reach CMMI-DEV maturity level 2. As a future work, we can suggest the possibility of defining a framework of agile techniques, practices and methodologies that could allow an organization to evolve into the five CMMI-DEV maturity levels. This framework could assure the institutionalization of agile practices into an organization, providing flexibility and quick response to changes as well as the use of a widespread model like CMMI, which guarantees higher quality processes and products.

7. ACKNOWLEDGMENTS

This research has been supported by the Tempros project (TIN2010-20057-C03-02) of the Ministry of Education and Science, Spain and by the NDTQ-Framework project (TIC-5789) of the Junta de Andalucía, Spain.

8. REFERENCES

- [1] Ran, H. et al. 2008. Agile Web Development with Web Framework. *In proceeding of 4th International Conference on Wireless Communications, Networking and Mobile Computing*. (China, October 12 – 17 2008) WiCOM '08. IEEE
- [2] Glazer, H. et al. CMMI or Agile: Why not embrace both!. Nov. 2008. *Carnegie Mellon University*. <http://www.sei.cmu.edu/reports/08tn003.pdf>. Last accessed 10-2012.
- [3] VersionOne. State of Agile Development Survey. 2011, from http://www.versionone.com/state_of_agile_development_survey/11/. Last accessed 10-2012.
- [4] CMMI Product Team. CMMI for Development, Version 1.3. Nov. 2010. *Carnegie Mellon University. Technical Report*. From <http://www.sei.cmu.edu/reports/10tr033.pdf> Last accessed 10-2012.
- [5] Herbleb, J. et al. Benefits of CMM-based software process improvement: Initial results. Aug. 1994. *Carnegie Mellon University*. <http://www.sei.cmu.edu/reports/94tr013.pdf>. Last accessed 10-2012.
- [6] Boehm, B., Turner, R. Balancing Agility and Discipline: A Guide for the Perplexed. 2003. *Addison-Wesley*.
- [7] Sutherland, J.; Schwaber, K. The Scrum guide: The Definitive Guide to Scrum: The Rules of the Game. 2011, from <http://www.scrum.org/Scrum-Guides>. Last accessed 10-2012.
- [8] Marçal, A. S. C. et al. Blending Scrum practices and CMMI Project management process areas. 2008. *ISSE* 4. 17-29.
- [9] Anderson, D. J. Stretching agile to fit CMMI level 3 – the story of creating MSF for CMMI process improvement at Microsoft Corporation. Jul. 2005. *In proceeding of Agile Development Conference 2005 (USA 24-29 July 2005)*.
- [10] Pikkarainen, M., Mäntyniemi, A. An approach for using CMMI in agile software development assessments: Experiences from three case studies. *In proceedings of Software Process Improvement and Capability determination 2006 Conference (Luxembourg 4-5 May 2006)*. SPICE '06.
- [11] Fritzsche, M., Keil, P. Agile methods and CMMI: Compatibility or conflict?. 2007. *e-Informatica Software Engineering Journal*. Volume 1, Issue 1.
- [12] Díaz, J. et al. Mapping CMMI Level 2 to Scrum practices: An experience report. Jan. 2009. *SPI* 42 93-104
- [13] Jakobsen, C. R., Johnson, K. A. Mature Agile with a twist of CMMI. Aug. 2008. *In proceeding of Agile Conference 2008 (Toronto, Canada 04-08 Aug. 2008)*. AGILE '08. IEEE.
- [14] Beck, K. et al. Manifesto for Agile Software Development. 2001, <http://www.agilemanifesto.org>. Last accessed 10-2012.
- [15] Schwaber, K. Scrum Development Process. Oct. 1995. *In proceedings of the 10th Annual ACM Conference on Object Oriented Programming Systems, Languages and Applications (Austin, TX, USA 15-19 Oct. 1995)*. OOPSLA '95. ACM.
- [16] Sutherland, J. et al. Scrum and CMMI Level 5: The Magic Potion for Code Warriors. Jan. 2008. *In proceeding of the 41st annual Hawaii International Conference on System Sciences 2008 (Hawaii, USA 07-10 Jan. 2008)*. HICSS '08. IEEE.
- [17] Kitchenham, B. et al. S. Systematic literature reviews in software engineering – A systematic literature review. 2009. *Information and Software Technology* 51 7–15.
- [18] Baker, S. W. Formalizing agility: An agile organization’s journey toward CMMI accreditation. Jul. 2005. *In proceeding of Agile Development Conference 2005 (Denver, CO, USA 24-29 July 2005)*. ADC '05. IEEE.
- [19] Baker, S. W. Formalizing Agility, Part 2: How an Agile Organization Embraced the CMMI. Jul. 2006. *In proceeding of Agile Conference 2006 (Minneapolis, USA 23-28 July 2006)*. AGILE '06. IEEE.
- [20] Cohan, S., Glazer, H. An agile development team’s quest for CMMI maturity level 5. Aug. 2009. *In proceeding of Agile Conference 2009 (Chicago, IL, USA 24-28 Aug. 2009)*. AGILE '09. IEEE.
- [21] McMahon, P. E. Integrating CMMI and Agile development. 2011. *Addison-Wesley*.
- [22] Cockburn, A., Highsmith, J. Agile software development: The people factor. Nov. 2001. *Computer* 11 131-133