Indian Journal of Science and Technology

Compatibility of Agile Software Development Methods and CMMI

Taghi Javdani Gandomani[1]* and HazuraZulzalil[2]

[1]Department of Computer Engineering, Boroujen Branch, Islamic Azad University, Boroujen, Iran; tjavdani@yahoo.com
[2]Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, University Putra Malaysia, Malaysia; hazura@fsktm.upm.edu.my

Abstract

Agile software development methods by focusing on the different values, have established a new approach for software development compare to the traditional methods. While traditional methods emphasizes on the following disciplined processes and rigid practices, agile methods focused on individuals and their collaborations. Previously, CMMI was introduced as process improvement model in the traditional methods and now it is known as symbol of those methods. Nevertheless, compatibility of these two different approaches has been a serious question and there are several conflicting reports about their compatibility. The main purpose of this study was scrutinizing this issue. Generic practices and specific practices in CMMI were selected as criteria for checking compatibility of CMMI and agile methods. The results showed that CMMI and agile are compatible in several process areas. However, there are serious incompatibilities in the others. In level 3 and level 4 of CMMI, two and one incompatible process areas have been seen respectively. Also, agile does not support most of the generic practices in CMMI level 4 and 5. The results also showed that these incompatibilities return to the notion of organization and its role in software development.

Keywords: Agile Methods, CMMI, Agile Software Development, Traditional Software Development.

1. Introduction

Agile methodologies as a reaction to traditional software development methodologies emphasize on the different values rather comparing to the traditional methods. After formal introduction of agile manifesto, several agile methods were gathered under agile umbrella. Though claiming to bring new achievements, software practitioners had and have serious concerns about replacing agile methods with traditional methods.

While traditional methods in software development have reached to an acceptable maturity level, lots of issues in agile methods are still open. At the same time agile values, tempt software practitioners to use agile methods as a replacement of traditional methods. Due to their different natures, agile and traditional methods have different activities, roles, practices and goals.

A wide range of companies and organizations rely on the well-known models like Capability Maturity Model Integration (CMMI) [27] for achieving higher quality and customer satisfaction [17, 19]. CMMI acts as indicator for organizational maturity and is widely used by companies that are using disciplined methods. In fact, CMMI is a symbol of traditional methods in software engineering. In fact CMMI has promised better quality of products through process improvement. Generally, it means that customers

Corresponding author:
Taghi Javdani Gandomani (tjavdani@yahoo.com)

have more confidence and trust in companies that have higher CMMI certification.

On the other hands, agile methods by relying on the people and their collaborations, try to decrease time to market and embrace changes even in the last stages. Furthermore, agile methods have promised early and frequent delivery, low ceremonies, lightweight documentation and working software. These make agile methods different from traditional methods. Such values motivate both customers and software companies to use agile methods for software development. In these days, it seems that agile values have more advocates. This is done while it seems that Agile approach and CMMI approach are not inconsistent, at least in their concepts.

Customers and organizations which are using CMMI as quality indicator and process improvement model are usually worried about compatibility of agile methods and CMMI. For answering to this concern they can find very different answers in previous studies and reports, from strongly incompatible to completely compatible. This study has tried to find a suitable and reliable answer to this question.

The rest of this article is formed as the following sections. The next section describes a brief background followed by Section 3 that explains research methodology. Section 4 discusses the results of the study and finally, Section 5 concludes the study and shows the potential future works.

2. Background

In this section, a brief background about CMMI is provided, followed by a short introduction of agile methods. Afterwards, previous related works are reviewed.

2.1 CMMI

CMMI and its predecessor CMM, in their various versions, have focused on process improvement in software organizations. Last version of CMMI for development, CMMI-dev V1.3, was published at 2010 [27]. In this model, four maturity levels are defined which each of them encompasses its own process areas (PA). Also, it defines several generic goals which should be passed in each maturity level. For reaching to a particular level of maturity, all of its PAs and generic practices should be achieved. Each PA has its own specific practices and goals. Thus, for reaching to a specific level of maturity, all of the associated specific practices should be institutionalized. Furthermore, generic practices in all PAs should be achieved.

2.2 Agile Methods

So far several agile methods are introduced which of them Scrum, Extreme Programming (XP), Test Driven Development (TDD), Feature Driven Development (FDD), Crystal family are more popular than others [11]. Although they have different practices, all of them follow same values. Between them, Scrum [12, 26], XP [6] and recently lean [24] and kanban [2] are mostly adopted [23, 31]. Since agile methods are people-centric, role of the individuals is more critical than processes [13]. These methods, in contrast to traditional methods, have focused on the new values such as light-weight documentation, embracing changes, iterative and frequent releases, working software, individuals and interactions, customer collaboration and so forth [6, 8, 26]. Although these methods were introduced formally at 2001, prevalence of them has increased almost after 2006. Several reputed companies and organizations now are using agile methods in some of their projects [9, 16, 21, 32]. It seems that companies need these methods to achieve new business benefits and therefore, agile adoption should be considered as a necessary process in software companies.

2.3 Related Works

By introducing agile methods, some of the software practitioners started writing about compatibility of agile methods and CMMI. Reviewing previous reports shows that, although early approaches were mostly negative, later they became more balanced.

Turner and Jain, after introduction of early versions of CMMI, compared it with agile manifesto and concluded that although there are some differences; “oil and water” description for CMMI and agile methods is overstated [30]. Boehm and Turner, believed that based on a defined risk analysis, organizations can decide using agile or plan-driven methods [7], However, they didn’t discuss on compatibility or incompatibility of these approaches. Fritzsche and Keil, studied compatibility of CMMI and Scrum and XP and concluded that CMMI practices can be adopted in these methods without any major conflicts in level 2 and 3, but there are major incompatibilities in level 4 and 5 [15]. Santana et al. showed that merging CMMI and agile methods is ignoring many important aspects of both approaches and it makes improvement so hard especially when merging process is performed in higher levels of agile manifesto and CMMI [25]. Cohan and Glazer, explained how their company moved from CMMI level 4 to level 5 while they added agile practices to their process [10].

Of course they had added several artifacts to achieving this goal, where these artifacts were not in agile approach. Sutherland et al in a CMMI level 5 company mixed Scrum and CMMI practices and discussed that their performance got better as compared with either Scrum or CMMI alone [29]. Glazer et al. in a detailed study for Software Engineering Institute (SEI), stated that both agile and CMMI can be used in software projects, However, they have differences in their approaches [28]. He later claimed that these different approaches need each other [18].

Several studies described the journey of agile adoption in CMMI organizations [1, 3, 4, 5, 9,14, 16, 20, 21, 22]. In most of them the main aim of researchers had been using both agile and CMMI approaches together, so, if there had been any conflicts, they had tried to use additional practices, documents and controls to overcome those incompatibilities. It is clear that such additional practices were not provided by original CMMI or agile methods. Table 1 shows the related works briefly.

To sum up, there were no common idea or finding on compatibility of agile and CMMI. It should be noted that most of the previous studies were conducted based on the conceptual understanding of agile, because there were not enough experiences on agile methods in real environments previously. This study has been done based on the real experiences on both agile and CMMI.

Table 1. provides some of the most important related studies.

Table 1. The most important related works

Author(s)	Year	Main findings
Turner and Jain	2002	Not too deep differences between CMMI and Agile
Boehm and Turner	2004	Proposing a risk-based plan for choosing Agile or CMMI
Fritzsche and Keil	2007	Major conflicts between Agile and CMMI level 4 and 5
Sutherland et al.	2007	Better performance by applying Scrum practices in a CMMI level 5 company
Santana et al.	2009	Merging Agile and CMMI leads to ignoring importance aspects of both them
Cohn and Glazer	2009	Empirical evidence of applying Agile in CMMI level 4 and 5
Glazer et al. (SEI)	2010	Differences of Agile and CMMI; Using both approach together
Glazer	2010	CMMI and Agile need each other

3. Research Methodology

This study through the popular professional on-line communities invited software practitioners to participate in this research. Having enough experience in both agile and CMMI was necessary requirement for attending this research. CMMI-Dev 1.3 as the latest version of CMMI was considered for benchmarking in this study. After providing a brief about the study, participants were asked about all 22 CMMI process areas and 16 generic practices through an online questionnaire. In each PA, compatibility of all its related specific practices with agile methods was asked. Next to each question, clear definition and description was provided too. They could answer by choosing one of these answers: Strongly Conflict (SC), Conflict (C), Neutral (N), Support (S), and Strongly Support (SS). Finally, collected data were analyzed.

4. Results and Discussion

During data collection period (two months), 41 answers were received. Most of the participants were from USA and west European countries. This study also collected detail information of the participants, but due to the space limitation, demography of participants is excluded from this article.

Since the questionnaire used 5-point Likert items, the first analysis was done on determining reliability of the questions or internal consistency. Using Cronbach's alpha test, showed that alpha coefficient for all questions was greater than 0.7 which showed the appropriate measure of reliability (generally around 0.8 to 0.85).

For each specific goal, its specific practices were asked. Considering agreement of participants associate to each PA specific practices, results are shown in Table 2. Also, findings of this study regarding to CMMI generic practices are demonstrated in Table 3. In both table percentage of frequency of the dominant view also is provided. Although the frequency only shows the opinion of the experts and so is subjective, for general judgment is helpful.

As results show, in CMMI level 2, Project Planning (PP) and Requirements Management (REQM) are supported by agile but other PAs are neutral to agile. In level 3, results show that while agile supports CMMI in most of the specific goals, it has clear conflict with CMMI in two process areas, Decision Analysis and Resolution (DAR) and Organizational Process Focus (OPF).

In CMMI level 4, although, Organizational Process Performance (OPF) is in conflict with agile, Quantitative

Table 2. Participants' agreement on CMMI process areas

CMMI level	Process Area	The Dominant View	Percentage
2	Measurement and Analysis (MA)	Neutral	60.3
2	Supplier Agreement Management (SAM)	Neutral	65.4
2	Configuration Management (CM)	Neutral	70.3
2	Project Planning (PP)	Support	66.2
2	Project Monitoring and Control (PMC)	Neutral	61.3
2	Process and Product Quality Assurance (PPQA)	Neutral	63.4
2	Requirements Management (REQM)	Support	69.8
3	Integrated Project Management (IPM)	Support	65.5
3	Decision Analysis and Resolution (DAR)	Conflict	55.2
3	Organizational Process Focus (OPF)	Conflict	55.3
3	Organizational Process Definition (OPD)	Support	54.7
3	Organizational Training (OT)	Support	55.7
3	Risk Management (RSKM)	Support	60.4
3	Requirements Development (RD)	Support	58.0
3	Product Integration (PI)	Support	59.0
3	Technical Solution (TS)	Support	48.3
3	Validation (VAL)	Support	49.6
3	Verification (VER)	Support	53.2
4	Organizational Process Performance (OPP)	Conflict	55.1
4	Quantitative Project Management (QPM)	Support	49.4
5	Organizational Performance Management (OPM)	Support	52.7
5	Causal Analysis and Resolution (CAR)	Neutral	44.8

Project Management (QPM) is supported by agile. Finally, in CMMI level 5, While Causal Analysis and Resolution (CAR) is natural to agile, Organizational Performance Management (OPM) is supported by it.

Reviewing Table 3 shows that most of the generic practices of CMMI level 2 are supported by agile. However, "Objectively Evaluate Adherence" seems to be in clear conflict with agile. Also, in higher levels more conflicts can be seen. Regarding to the generic practices of CMMI level 3, while one of them, "Establish a Defined Process" is supported by agile, the other one, "Collect Improvement Information" is in conflict with it. Furthermore, both of two generic practices of CMMI level 4 are seen in conflict with agile. Finally, in CMMI level 5, like level 3, one generic practice, "Ensure Continuous Process Improvement" is supported by agile but the other one, "Correct Root Causes of Problems" is in conflict with it.

Based on the above results, general judgment on compatibility of CMMI and agile is debated. While in one level, there is no conflict, in another level clear conflicts can be seen. The results show that wherever, CMMI stresses on organization, agile has conflict with it. In the other words, the major conflicts between two approaches return to the organization notion.

5. Conclusion and Future Works

CMMI and agile focus on software process from different perspectives. While CMMI focuses on rigid, predictable, well-documented and plan-driven process, agile stresses on individual collaboration, embracing change and lightweight ceremonies. Both of two approaches are attractive and useful on their own. The main issue is compatibility of these approaches.

Conducting a structured survey, showed that, general judgment about the compatibility of these approaches is not easy. Considering both generic practices and process areas, results showed that while agile is supportive and neutral

Table 3. Participants' agreement on CMMI generic practices

CMMI level	Generic Practice	The Dominant View	Percentage
2	Establish an Organizational Policy	Support	65.5
2	Plan the Process	Support	51.9
2	Provide Resources	Support	67.3
2	Assign Responsibility	Support	70.1
2	Train People	Support	66.2
2	Manage Configurations	Support	63.5
2	Identify and Involve Relevant Stakeholders	Support	70.8
2	Monitor and Control the Process	Neutral	53.6
2	Objectively Evaluate Adherence	Conflict	59.2
2	Review Status with Higher Level Management	Neutral	63.9
3	Establish a Defined Process	Support	71.3
3	Collect Improvement Information	Conflict	46.2
4	Establish Quantitative Objectives for the Process	Conflict	58.6
4	Stabilize Sub process Performance	Conflict	60.3
5	Ensure Continuous Process Improvement	Support	57.4
5	Correct Root Causes of Problems	Conflict	55.0

in lower levels, there are major conflicts in higher levels. Especially when CMMI stresses on the notion of organization, agile is in conflict with it.

This study did not focused on reasons of incompatibilities and on how to remove the incompatibilities. Both of these subjects can be studied in separate research as future works of this study.

6. Acknowledgement

This study was performed as a research plan and was financially supported by Islamic Azad University, Boroujen Branch, Iran.

7. References

1. Anderson D J (2005). Stretching agile to fit CMMI level 3: The story of creating MSF for CMMI process improvement at Microsoft Corporation, Paper presented at the AGILE Conference 2005, Denver, CO.
2. Anderson D J (2012). Lessons in Agile Management: On the Road to Kanban, Blue Hole Press, USA.
3. Babuscio J (2009). How the FBI learned to catch bad guys one iteration at a time, Paper presented at the Agile Conference, AGILE 2009, Chicago, IL.
4. Baker S W (2005). Formalizing agility: An agile organization's journey toward CMMI accreditation, Denver, CO, 185–192.
5. Baker S W (2006). Formalizing agility, part 2: How an agile organization embraced the CMMI, Minneapolis, MN, 154.
6. Beck K and Andres C (2004). Extreme Programming Explained: Embrace Change, 2nd Edn., Addison-Wesley Professional, Boston.
7. Boehm B, and Turner R (2004). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods, 26th International Conference on Software Engineering, ICSE 2004, Edinburgh, 718–719.
8. Chromatic (2013). Extreme Programming Pocket Guide, 1st Edn., O'Reilly Media, Sebastopol, CA.
9. Chung M W, and Drummond B (2009). Agile @ yahoo! from the trenches, Paper presented at the Agile Conference, AGILE 2009, Chicago, IL, 113–118.
10. Cohan S and Glazer H (2009). An agile development team's quest for CMMI® maturity level 5, Paper presented at the Agile Conference, AGILE 2009, Chicago, IL, 201–206.
11. Cohen D, Lindvall M et al. (2004). An Introduction to Agile Methods, Advances in computers, vol 62, 1–66, doi: 10.1016/S0065-2458(03)62001-2.
12. Cohn M (2009). Succeeding with Agile: Software Development using Scrum, 1st Edn., Addison-Wesley Professional, Boston, MA.

13. Conboy K, Coyle S et al. (2011). People over process: Key challenges in agile development, IEEE Software, vol 28(4), 48–57, doi: 10.1109/ms.2010.132.
14. Diaz J, Garbajosa J et al. (2009). Mapping CMMI level 2 to scrum practices: An experience report, Communications in Computer and Information Science, vol 42, 93–104.
15. Fritzsche M and Keil P (2007). Agile Methods and CMMI: Compatibility or Conflict? e-Informatica Software Engineering, vol 1(1), 9–26.
16. Fulgham C, Johnson J et al. (2011). The FBI gets agile, IT Professional, vol 13(5), 57–59, doi: 10.1109/mitp.2011.88.
17. Galin D, and Avrahami M (2006). Are CMM Program Investments Beneficial? Analyzing Past Studies, Software, IEEE, vol 23(6), 81–87, doi: 10.1109/ms.2006.149.
18. Glazer H (2010). Love and marriage: CMMI and agile need each other, CrossTalk, vol 23(1–2), 29–34.
19. Goldenson D R, and Gibson D L (2003). Demonstrating the Impact and Benefits of CMMI (1 ed., Vol. CMU/SEI-2003-SR-009, pp. 1–38). Pittsburgh, PA: Software Engineering Isntitute (SEI).
20. Laanti M (2010). Agile transformation study at Nokia - One year after, Paper presented at the 1st International Conference on Lean Enterprise Software and Systems, LESS 2010, Helsinki.
21. Laanti M, Salo O et al. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. Information and Software Technology,vol 53(3), 276–290, doi: 10.1016/j.infsof.2010.11.010.
22. Marçal A S C, De Freitas B C C et al. (2007). Mapping CMMI project management process areas to SCRUM practices, SEW '07 Proceedings of the 31st IEEE Software Engineering Workshop, 13–22, Columbia, MD.
23. Pikkarainen M, Haikara J et al. (2008). The impact of agile practices on communication in software development, Empirical Software Engineering, vol 13(3), 303–337, doi: 10.1007/s10664-008-9065-9.
24. Poppendieck M and Poppendieck T (2009). Leading Lean Software Development: Results Are not the Point, Addison-Wesley Professional, Boston, MA.
25. Santana C, Gusmão C et al. (2009). Agile software development and CMMI: What we do not know about dancing with elephants, Paper presented at the 10th International Conference on Agile Processes in Software Engineering and Extreme Programming, XP 2009, Pula, Sardinia, 124–129.
26. Schwaber K (2007). The Enterprise and Scrum, 1st Edn., Microsoft Press, Redmond, Washington.
27. SEI (2010a). CMMI-Dev 1.3, Carnegie Mellon University: Software Engineering Institute (SEI).
28. SEI (2010b). CMMI or Agile: Why Not Embrace Both, Pittsburgh, PA: Software Engineering Institute (SEI).
29. Sutherland J, Jakobsen C R et al. (2007). Scrum and CMMI level 5: The magic potion for code warriors, Big Island, HI.
30. Turner, R., & Jain, A. (2002). Agile Meets CMMI: Culture Clash or Common Cause? Paper presented at the Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe 2002, vol 2418, 153–165.
31. VersionOne. State of Agile Survey 2011, Available from http://www.versionone.com/state_of_agile_development_survey/11/
32. Williams L, Brown G et al. (2011). Scrum + engineering practices: Experiences of three microsoft teams, Paper presented at the 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011, Banff, AB.