

Scrum and CMMI: A High level assessment of compatibility

Srinivas Chillara¹ and Pete Deemer²

Abstract:

This article's purpose is to assess the compatibility of Scrum with CMMI and also provide a base for understanding how Scrum can be implemented in organizations that are following a CMMI model. To an extent this is also a comparative study. Introductions to Scrum and CMMI are provided, followed by a discussion on the compatibility of Scrum and CMMI, at an implementation and an approach level.

0. INTRODUCTION

CMMI is a process framework which is widely adopted by software and systems development companies, while Scrum is one of the most important software development methodologies whose adoption is growing rapidly. CMMI is basically a process improvement framework which mandates a set of processes for software development management. Scrum can be thought of as an iterative software project management framework for development activities. CMMI has a wider scope and different aims to those of Scrum. CMMI also covers production-support, maintenance, product implementation and application transition type projects as well.

1. SCRUM OVERVIEW

Scrum is an iterative, incremental framework for managing projects, including software development projects. Scrum's approach to control software development is *empirical*, since it is known to be highly complex with many factors, some human dependant. Consequently, it is difficult to measure or predict the development effort and results. Scrum structures work in iterations called Sprints, which are of four weeks in length and take place one after the other for the duration of a project. At the beginning of each Sprint, a cross-functional team selects items from a prioritized list of requirements called the Product Backlog; this list is created and maintained by the Product Owner, who gathers inputs from customers and others, and then ranks them according to business value. The team selects items starting at the top of the Product Backlog (in other words, the highest business value items), and they only select as many as they believe they can complete at high quality by the end of the Sprint. Scrum places a very strong emphasis on producing a potentially shippable increment of product at the end of each Sprint; in the case of software, this means code that has been not only written, but also fully integrated, tested, documented, and any other steps necessary to reach "done". Once the team has made its commitment, there is no change during the Sprint, either to the

¹ Independent consultant, CSM. Srinivas has over 10 years experience in software development and management. He has a B.Tech from IIT Kharagpur, India and an MSc from University of Reading, UK. He can be reached at; ceezone@yahoo.co.in

² Primary consultant at Goodagile, CST. Pete has over 16 years of experience. He has been a product director of Yahoo Bangalore and co-lead the successful large scale adoption of Scrum there. He is a graduate of Harvard University. He can be reached at: petedeemer@goodagile.com

³ In this article we are only considering the "staged representation" of the CMMI model because it is simpler, as well familiar to the authors.

deliverable (any new requests have to wait until the next Sprint), or to the length of the Sprint (which ends on the pre-determined date, whether or not the work has been completed). During the Sprint, the team manages itself to complete what it committed to; as part of this, the team gathers daily for 15 minutes at the Daily Scrum Meeting, to update each other on progress, surface impediments, and update Scrum artifacts such as the Sprint Backlog (the tasks that comprise the deliverable), and the Sprint Burndown Chart (a graph of the hours remaining to complete the deliverable). At the end of the Sprint, the team holds a Sprint Review, where they demonstrate what they have built, to generate feedback which the Product Owner can incorporate into the Product Backlog for later Sprints. The team also holds a Sprint Retrospective, where they review their way of working, and agree on changes to improve their effectiveness. Each Sprint is self-contained; before the start of the next Sprint, the Product Owner can make whatever changes he/she likes to the remaining Product Backlog, adding, removing or reprioritizing items at will. One critical new role for a team's success with Scrum is: the ScrumMaster. The ScrumMaster is not the team's manager; instead, he/she protects the team from outside interference or disruption, serves the team by helping get impediments removed. He/she also facilitates team interactions, as the team and organization is coached in the use of Scrum.

Many people mistakenly perceive Agile methods to be loose or sloppy; however, when practiced as intended, Scrum is very strict and disciplined, with a great emphasis on transparency, commitment, accountability, quality, and continuous improvement. Scrum is used mostly by product development companies.

2. CMMI OVERVIEW

CMMI is a process improvement framework for software systems development companies. As Judy Bamberger (a key author of the CMM) says [“The essence of the CMM”] “The CMM wasn’t intended to be all things to all people to cover all possible aspects of software and system development. It was intended to provide a set of guidelines for managing software development projects and making improvements over time”. This framework is mostly at a management level and seldom addresses core engineering issues in depth. The motivation for adoption of CMMI is intended to bring about a higher software development maturity of the organization, which in turn results in smoother, more efficient lower defect deliveries.

CMMI consists of a number of "Practice Areas" (PA), which are grouped into four maturity levels that organizations should go through before having a process improvement system in place. The maturity levels are:

1. Initial (ad hoc, heroic) (no PA)
2. Managed (project management, process discipline)
3. Defined (institutionalized)

4. Quantitatively Managed (quantified)

5. Optimized (process improvement)

One way companies are supposed to use the model is first to assess their maturity level and then form a specific plan to get to the next level. The SEI has defined a rigorous process assessment method to appraise how well an organization satisfies the goals associated with each level. Skipping levels is not allowed.

Maturity Level	Name	Degree of Scrum compliance (and how)
2	Requirements Management	High (PB, RB, SB)
2	Project Monitoring and Control	High (PB, RB, SB and Daily scrums)
2	Project Planning	High (SPM)
2	Supplier Agreement Management	Not applicable
2	Configuration Management	Not Addressed
2	Measurement and Analysis	Medium (SPM, SB, burndown chart)
2	Process and Product Quality Assurance	High (SRM)
3	Product Integration	High (Sprints and SRM)
3	Requirements Development	Medium (SPM,SRM)
3	Technical Solution	Not Addressed
3	Validation	Medium (frequent SRM)
3	Verification	High (“done” criteria and

		SRM)
3	Organizational Process Definition	Low (Scrum process itself)
3	Organizational Process Focus	Low (Scrum process itself)
3	Organizational Training	Not addressed
3	Integrated Project Management	High (SRM, Scrum of Scrums and PB)
3	Integrated Supplier Management	Not applicable
3	Integrated Teaming	High (Daily Scrum, Scrum of Scrum)
3	Risk Management	Medium (Iterative approach limits risk)
3	Decision Analysis and Resolution	Not addresses
3	Organizational Environment for Integration	Medium (Scrum of Scrums)
4	Organizational Process Performance	Not addressed
4	Quantitative Project Management	Low (Estimation during SRM and burndown chart)
5	Organizational Innovation and Deployment	Not applicable
5	Causal Analysis and Resolution	High (daily Scrums and Retrospectives)

Table 1: CMMI PA levels and extent to which Scrum satisfies these
NA: Not Applicable, because this is usually not of interest.

Most organizations which follow CMMI are service companies; relatively much fewer product development companies adopt CMMI formally or otherwise. Many organizations also obtain a CMMI (level 3 and above) certification⁴ since this opens doors to the tenders floated by many large US corporations for their IT/IS projects. Vendor parties which don't have such a certification are not even short listed.

3. IS SCRUM COMPATIBLE WITH CMMI?

Yes and no. Since CMMI mandates certain PAs, but not how these PAs should be satisfied; Scrum practices can and do meet many of these PAs (see table 1). So at an implementation level Scrum and CMMI are mostly compatible, and following Scrum properly will meet a majority of CMMI PAs. A good exposition of the mapping of Scrum practices to CMMI project management PAs is provided in [Marcal et al]

However CMMI is mainly intended to be a process improvement framework while Scrum is a lightweight framework for helping a project group focus efforts of the team towards achieving project goals.

Moreover there are two underlying assumptions of CMMI which are rejected by Scrum.

- 1) Software development processes are(or can be) defined (level 3 is labeled: **Defined**).
- 2) A good set of processes is the primary means to deliver software efficiently.

Scrum favours an approach where the software development process is taken to be empirical. Further Scrum believes that people, their skills and how they are harnessed are the most important factors for the success of a software development effort. Due to these differences in approach, the spirit that CMMI fosters is considerably different from that of Scrum.

Comments on how Scrum satisfies PAs at different levels follow:

At level2: Scrum takes care of most PAs via various levels of backlogs (which are feature lists), sprint planning meetings, daily scrum, burn-down chart and sprint review meeting. Since Scrum does not mandate engineering approach, it does not comment on configuration management, hence this PA is not addressed.

At level 3⁵: This is critical CMMI level with a number of PAs, many of which are met by Scrum. SRM, daily Scrum, regular updating of sprint backlog and 'scrum of scrums' for larger project teams satisfy these PAs. Scrum does not comment on technical solution or on Decision analysis and resolution (DAR). The other PAs which Scrum does not address satisfactorily are

⁴ In reality there is nothing like a CMMI "certificate" handed out by SEI.

⁵ It is said that an organization must at least achieve level4 to reap the benefits of CMMI; this is because process improvement doesn't kick in, unless one starts implementing level4. One can find many organizations at level3 and some at level5. Few companies reach level4 and stop there for long, for these have a motivation to progress onwards to level5.

to do with organizational standardization. On reflection this is only natural since the aim of Scrum is to manage a given project to deliver working software, and that of CMMI is the have an organizational project execution capability in place.

At level4: Scrum fares badly from a CMMI perspective. Scrum does not stress a quantitative approach (i.e. metrics based) to project management. The burn-down chart provides a velocity metric which partly addresses this; however this is not intended to be of organization wide scope. However Scrum practices don't forbid collection and analysis of metrics, unless they turn out to be impediments to the team's work.

At level5: Scrum has an important practice of Sprint Retrospectives which is a key manner in which "inspect and adapt" is done. Implemented effectively this is very useful process monitoring and control method which largely meets the causal analysis and resolution PA.

4. CONCLUSION

Overall the main CMMI PAs which Scrum does not cover, are to do with organizational standardization, which is a CMMI aim, not a Scrum concern. Scrum practices are as such compatible with CMMI framework. Scrum practices can therefore form a large part of the CMMI implementation. However a faithful implementation which maintains the spirit of Scrum is fraught with difficulties. There are two very important differences in the underlying assumptions of Scrum compared to CMMI: 1. Software development is empirical 2. People are more important than processes.

To adopt Scrum well in a CMMI organization, someone with a good understanding of both CMMI and Scrum must be available to guide the team and the organization. Given the right circumstances, particularly upper management support, with such guidance it is possible to implement Scrum well within a CMMI organization; essentially this would mean implementing CMMI via Scrum for the development projects or a sub-set, and then filling the gaps such that the spirit of Scrum isn't disturbed.

REFERENCES

- [1] Craig Larman and Victor Basili., 2003 IEEE computer, "Iterative and Incremental Development: a brief history"
- [2] Mike Cohen and Doris Ford., 2003 IEEE computer, "Introducing an Agile process to an organization"
- [3] Ana Sofia Marcal, Bruno de Freitas, Felipe Soares and Arnaldo Belchoir, "Mapping CMMI project management process areas to SCRUM practices" (internet)

