A case study of software process improvement with CMMI-DEV and Scrum in Spanish companies

Javier Garzás1,* and Mark C. Paulk2

1Kybele Research and Kybele Consulting, Rey Juan Carlos University, Tulipán s/n, Móstoles, Madrid, Spain  
2Institute for Software Research, Carnegie Mellon University, 5000 Forbes Avenue, Wean Hall 5101, Pittsburgh, PA, 15213, USA

ABSTRACT

One of the most commonly used agile methods is Scrum. Capability Maturity Model Integration for Development (CMMI-DEV) is currently the de facto framework for process improvement and for determining the organizational maturity of software development companies. CMMI-DEV and Scrum share certain characteristics, and even though they were developed for different purposes, they can be complementary to each other; and as such, they are not in competition. This paper presents a case study of the relationship between level 2 of CMMI-DEV 1.3 and Scrum. This research has focused on the relationships between Scrum and level 2 of CMMI-DEV 1.3. The objective of this research paper is to evaluate how Scrum helps implement a process model such as CMMI-DEV.

A detailed case study was conducted among Spanish IT companies. The case study was designed according to established guidelines for case studies. There were eight principal activities: case study design, case selection, case study procedures and roles, data collection, analysis, plan validity, study limitations, and reporting.

The results obtained show that most of the process areas of CMMI-DEV level 2 had been improved by using Scrum. Other issues detected arose during the formal appraisals and illustrated how it is possible to verify with Scrum that the specific goals of CMMI-DEV have been implemented. In addition, it highlights how the use of open-source tools was useful in improving the process in the companies involved. Based on the case study carried out, the addition of Scrum methodologies may improve the increase in quality of software processes. Copyright © 2013 John Wiley & Sons, Ltd.

1. INTRODUCTION

The lack of empirical validations is one of the reasons for the gap between practice and research [1]. Empirical validation methods for software engineering are useful because they reflect real experience. However, as Glass et al. [2] comment, in software engineering, there are, unfortunately, very few studies that use research or empirical validation methods.

Zelkowitz [3] argues that empirical validation methods may be broken down into 11 types (Table I), and among them the most commonly used in software engineering is the case study. Increasingly, cases studies are necessary to confirm the validity of the application of software engineering practices [4]. A case study ‘collects detailed project data to determine if the developed product is easier to produce than similar projects in the past’ [3]. In a case study, the projects are monitored and ‘observational’ methods are used to gather relevant data during the life cycle of the study [5]. These real experiences can be used to improve software engineering practices.

*Correspondence to: Javier Garzás, Kybele Research and Kybele Consulting, Rey Juan Carlos University, Tulipán s/n, Móstoles, Madrid, Spain.
†E-mail: javier.garzas@urjc.es

Copyright © 2013 John Wiley & Sons, Ltd.
One of the most popular agile methods is Scrum [6]. Scrum can be described as a project management methodology based on using an iterative and incremental framework for software development. Scrum can be characterized by a relatively small set of agile practices, but Scrum is frequently adapted with other agile practices or methods such as Extreme Programming (XP) [7]. Cockburn [8] discusses a number of what he calls ‘sweet spots’ for agile projects, such as two to eight people in one room, on-site usage experts, 1-month increments, fully automated regression tests, and experienced developers.

Capability Maturity Model Integration for Development (CMMI-DEV) is currently the de facto framework for process improvement and for determining the organizational maturity of software development organizations [9]. Organizations can improve productivity, quality, and predictability by attaining ‘levels’ of CMMI-DEV.

The CMMI-DEV and Scrum share certain characteristics, even though they were developed for different purposes. They focus on different levels of abstraction: CMMI-DEV focuses on what organizations do, including their projects, and Scrum focuses on how projects develop products. As Glazer et al. [9] argue, Scrum provides a software project management how-to that is missing in CMMI-DEV, and CMMI-DEV provides the process management to continuously improve the use of Scrum. CMMI-DEV outlines ‘what to do’ but it does not explain ‘how to do’ it. Agile methods, such as Scrum, are best practices that contain a how-to for a particular type of environment [9, 10].

This research focuses on the relationships between Scrum and level 2 of CMMI-DEV 1.3. In this paper, we will present a case study on the relationship between CMMI-DEV and Scrum. We conducted this case study following the template of Brereton et al. [11] and guidelines set out in [12–14]. The following subsections describe the study. An overview of related works is shown in Section 2. Section 3 shows the research design. Section 4 describes the discussion and analysis of results. Section 5 outlines the validity and limitations of this research. Finally, we will present the conclusions and plans for future work in Section 6.

2. RELATED WORKS

For some time now, there have been several initiatives specifically oriented at putting together processes and agile methods, such as XP [15, 16]. Different studies present mappings between agile practices and software development process models such as CMMI-DEV [17–19] or standards such as ISO/IEC 12207 [20]. Authors such as Glazer et al. [9] and Johnson and Sutherland [21] argue that CMMI-DEV and agile methods can work together, because they work at different levels of abstraction, CMMI-DEV sets out what to do, whereas agile methods are guidelines on how to do software development.

In the case of Scrum, Sutherland et al. [22] and Jakobsen et al. [23] discuss the relationships between CMMI-DEV level 5, high level maturity, and Scrum. Other authors, such as Marcal et al. [24] identify the relationships between CMMI-DEV and Scrum. In [25], the author show how CMMI-DEV practices for the Measurement and Analysis (MA) Process Area can be introduced without harming the agility of a Scrum software development process. Similar studies, for example,
Jakobsen and Sutherland [26] and Potter and Sakry [27], identify equivalences between CMMI-DEV and Scrum. The last two aforementioned works identify equivalences between CMMI-DEV process areas and Scrum for related ‘specific practices’ of the process areas of maturity levels 2 and 3.

On the other hand, Pino et al. [28] present a similar case study about the Competisoft model that does not specifically address Scrum companies. There are a number of case studies on agile methods or Scrum, and how they are implemented in companies (some examples are [29–34] or [35]). There are only a few case studies about the relationships between Scrum and CMMI-DEV, one example of which is [36].

3. RESEARCH DESIGN (DESIGN, SUBJECTS, AND ANALYSIS UNIT)

A case study explores a phenomenon within its real context [13]. Runeson et al. [37] identified the case study as often being a good method for research in software engineering. This case study has been designed according to the guidelines for case studies presented by Brereton [11]. There are eight main activities: case study design, case selection, case study procedures and roles, data collection, analysis, plan validity, study limitations, and reporting.

3.1. Case study design

According to Yin [13], the design type of this case study is multiple cases – holistic, because the strategy has been applied in the context of 12 companies.

In this case study, all companies prior to agreeing to take part in the study had been using Scrum methods, and they had started a program of software process improvement.

The main research question addressed by this study is as follows:

• Is Scrum useful and practical for carrying out software process improvement efforts with models such as CMMI-DEV?

With this question, we sought to discover whether Scrum has a useful function in software process improvement in CMMI-DEV implementations. The scope of the study is limited to the level 2 process areas in CMMI-DEV, which are the following (Figure 1):

• Requirements Management (REQM)
• Project Planning (PP)
• Project Monitoring and Control (PMC)
• Configuration Management (CM)

Figure 1. Number of absent, weak, or erroneous implementations per specific goals. SG, specific goal; REQM, Requirements Management; PP, Project Planning; PMC, Project Monitoring and Control; CM, Configuration Management; MA, Measurement and Analysis; PPQA, Process and Product Quality Assurance.
• Measurement and Analysis (MA)
• Process and Product Quality Assurance (PPQA)
• Supplier Agreement Management (SAM)

The ‘SAM’ was out of scope because it did not apply to any of the companies in this case study. None of the companies had an external provider to develop part of their products. SAM is therefore considered ‘not applicable’.

3.2. Research subjects and analysis unit

The following criteria for research subject selection were used. Companies had to want to improve their software development process and be willing to work with Scrum. Subsequent to this, we studied 12 companies from Spain.

Some companies were using level 2 of CMMI-DEV as a reference model, or similar process models that addressed the same goals. In this study, the status of each company has been mapped to level 2 of CMMI-DEV 1.3.

All companies were given training for at least 6 months before this assessment. The main topics covered during the training were software processes and software process improvement models. The training also covered both an introduction to CMMI-DEV and Scrum.

The size of the development team varied between 6 and 52 people, as can be seen in Table II. The companies organize into team sizes of $7 \pm 2$ and most of them end up with multiple team projects. The products and services offered included Web technology, telecommunications, embedded software, and general purpose development. In all companies, the development was performed in-house. To preserve confidentiality, we will refer to participants by the designations C1 to C12.

3.3. Field procedure, data collection, and limitations

A 2-day visit was made to each company to assess the development of the processes. The procedure to extract the information and the data collection from the case studies was not a formal appraisal, but it was based on CMMI-DEV/SCAMPI B/ISO 15504–2. In each assessment, the assessment team studied the software development department and its improvement opportunities. We collected data using semi-structured interviews. We used a structured questionnaire.

Table III summarizes the information obtained by analyzing and summarizing the data collected on each company. The first two columns contain process areas and specific goals for CMMI-DEV level 2. In Table III an ‘X’ value is assigned in some of the following situations:

• Absent or poorly addressed implemented specific goals.
• Little or no evidence of achievement of some of the specific goals.

Table II. Companies description.

<table>
<thead>
<tr>
<th>ID</th>
<th>IT employees</th>
<th>Years</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>20</td>
<td>8</td>
<td>Telecommunications</td>
</tr>
<tr>
<td>C2</td>
<td>20</td>
<td>11</td>
<td>Product development</td>
</tr>
<tr>
<td>C3</td>
<td>52</td>
<td>9</td>
<td>General purpose</td>
</tr>
<tr>
<td>C4</td>
<td>6</td>
<td>15</td>
<td>General purpose</td>
</tr>
<tr>
<td>C5</td>
<td>20</td>
<td>7</td>
<td>Consulting and engineering</td>
</tr>
<tr>
<td>C6</td>
<td>30</td>
<td>9</td>
<td>Security</td>
</tr>
<tr>
<td>C7</td>
<td>6</td>
<td>8</td>
<td>Web</td>
</tr>
<tr>
<td>C8</td>
<td>10</td>
<td>6</td>
<td>ERP/DB (Enterprise Resource Planning and database)</td>
</tr>
<tr>
<td>C9</td>
<td>20</td>
<td>15</td>
<td>Embedded software</td>
</tr>
<tr>
<td>C10</td>
<td>15</td>
<td>6</td>
<td>Telecommunications</td>
</tr>
<tr>
<td>C11</td>
<td>7</td>
<td>6</td>
<td>ERP/DB</td>
</tr>
<tr>
<td>C12</td>
<td>20</td>
<td>5</td>
<td>General purpose</td>
</tr>
</tbody>
</table>

Employees, total number of employees involved in software development; years, number of years the company has existed; field, commercial scope of the products developed by the organization.
Some gaps or issues were identified that will prevent goal achievement or which might threaten goal achievement.

Specific goals that were only partially addressed in the set of implemented practices.

Some aspects of achievement of the specific goals may be unpredictable.

Previous cases have been extracted from the ‘red’ and ‘yellow’ labels of the colors scale described in [38] and from ‘N’ and ‘P’ attributes of the rating scale [39].

In Table III, the data in the bar chart shows the total number of ‘X’, that is, the absent, weak, or erroneous implementations per specific goals.

4. DISCUSSION AND ANALYSIS OF RESULTS

In this section, we will highlight the most relevant aspects in the 12 organizations.

In companies where any issues related to REQM arose (75%), these came from how to ‘ensure alignment between project work and requirements’, the ‘nonfunctional requirements specification’ or the ‘user acceptance test’ [6]. Although CMMI-DEV does not explicitly require testing software or acceptance tests to ensure alignment between project work and requirements, that is, the common practice for verification and validation, which are explicitly contained in CMMI-DEV. Testing is not among the explicit practices of Scrum, although test-driven development and continuous integration are common agile practices. Issues might arise to the degree that neither Scrum nor CMMI-DEV explicitly drives software testing.

No special issues were detected in relation to CM (only one company had some problems). Although Scrum does not cover the configuration management area, most companies have implanted the ‘continuous integration’ practice, which implies CM. This practice, and the use of software management tools, helped many of the companies.

Regarding the aforementioned topics of REQM and CM, most companies (91.67%, all except one) used open-source tools. To improve the software process and to use Scrum, most of the companies highlighted two particular types of tools: the bug (or issue tracking) system and the version control tool. The bug (or issue tracking) system was used for many project management tasks. Most
companies made adjustments and adaptations to the issue tracking system that aligned with the requirements of CMMI-DEV or software process improvement. For project management tasks and bug (or issue tracking) system, the tool most often used was Trac. For control version, the tool most often used was Subversion. Most companies used these kinds of tools to achieve bidirectional traceability. Bidirectional traceability is a recommended practice to implement the REQM process of CMMI-DEV. The link between Trac and Subversion is very common among companies, and this link between tools is useful to achieve bidirectional traceability. This type of solution was also useful to improve and manage requirements changes effectively, change the history or the impact of changes.

Concerning project management (PP and PMC), a recurrent issue is the lack of an established estimation method (83.3%). Most companies did not use any estimation method, and ‘expert judgment’ was the most common way to estimate. Curiously, ‘planning poker’ was rarely observed. Although agile methods and Scrum do not cover process areas such as Risk Management, this area was correctly implemented in most of cases. There are many ways to implement it; however, the companies addressed the basic practices of Risk Management in CMMI-DEV.

As to MA metrics, most companies collected some type of software measurement; however, in most cases, there was no relationship between these software metrics and the business objectives or information needs (41.7%). According to CMMI-DEV 1.3, the best practice is that measurement objectives document the purposes for which measurement and analysis are performed. In the case study, the majority did not reflect this best management practice (66.7%). The management of the software measurement was very informal, and the Chief Information Officer was rarely involved with the measurement results. On the other hand, most of the companies collected ‘white box metrics’ because it was easy to obtain with several open-source tools, such as PMD or CheckStyle. The second most common kind of metrics was related to project monitoring: effort, budget, and so on. And finally, the metrics related to black-box testing.

Regarding PPQA, sprint retrospectives were very useful. The aim of this agile practice is continuous improvement. The main problem in this process comes from a possible lack of objectivity. In most companies, it was difficult to find reviewers who were (a) not related to the project and (b) fully objective.

Generic goals (GG) describe activities that address aspects of institutionalization. Concretely, at level 1, ‘GG 1 achieve specific goals’ address that the ‘specific goals of the process areas are supported by the process by transforming identifiable input work products into identifiable output work products’ [40]. There were no specific issues with GG 1. At level 2, ‘GG 2 institutionalize a managed process’ establishes that the processes are institutionalized as a managed process. The main issues came from establishing an organizational policy (75%) at level 3.

On the other hand, all of the companies (100%) argued that adding software process improvements to their agile practices had been valuable. Some benefits mentioned were the generation of historic data, and to have a more real and generic vision of the organization, improving the monitoring, being better prepared to grow, competitive advantage, and so on.

All of the companies (100%) had some Scrum practices that did not leave a paper trail that could retrospectively help to demonstrate the implementation of some process areas or specific goals of CMMI. In these cases, a meeting was held with the person responsible (Scrum Master or similar role) for the processes to be appraised. Software companies may need to add artifacts to their Scrum practices to prepare a formal appraisal, as this was a recurring issue in all assessments. In this regard, most companies expressed concern about the formal appraisals and paper evidence (or the need to have evidence in a reproducible format); some companies even said that they would not conduct a formal appraisal due to the cost of preparing the paper evidence. In many cases, the creation of evidence, only for a formal appraisal and to justify that a specific goal has been reached, took much effort and added no value.

---

Problems with the estimation method could affect the monitoring of the project. However, these problems have been counted only in the PP process area.
5. VALIDITY AND LIMITATIONS

To address threats to the validity of the case studies, the following issues were considered:

- The design of the case study and the data collection plan were checked against the checklists for case studies proposed by Host et al. [14].
- Regarding the construct validity of the case study, we collected information from participants with different roles. Usually, we interviewed some developers, quality managers, Scrum Masters or project managers, and so on.
- As to the internal validity, the case study presented shows that the decision to use Scrum to guide process improvement allowed them to obtain key improvements. In the perception of the stakeholders, there was a causal relationship between adopting Scrum and improving process performance.
- Respecting the external validity, we had already implemented this type of study with other companies and similar models such as ISO 15504, or without agile practices. This allowed validating and refining of these case studies. Generalizing these observations to other organizations, integrating CMMI-DEV and agile methods seems reasonable, and we observe no issues that are unique to the context of this case study.

The case studies carried out have some limits:

- The observations and conclusions presented are based on 12 case studies, which can limit the power of generalization. Although these companies are representative of the software industry in Spain. We recognize that the number of companies taking part in the case studies is a low number, although arguably sufficient to draw general conclusions.
- Not all companies had the objective of obtaining an official CMMI-DEV certification. This could have varied the number of artifacts or evidence that companies were willing to generate. In particular, this affects the degree of explicit documentation, which may not add value in the organization’s eyes, that will be created or maintained, even though such documentation may be necessary to substantiate process capability in formal process appraisals.
- As Ken Schwaber [41] argued several changes, or cultural shifts, are required to use Scrum. We should consider that the cultural differences between countries may have affected this study. The adoption and means of implementing Scrum could be different in other countries. National and ethnic cultures, which are intrinsic to the workforce, may be expected to affect organizational cultures, which are acquired and may be more easily (if not necessarily easily!) changed.

6. CONCLUSIONS

In this article, a case study of Spanish companies and the relationships between CMMI-DEV process and Scrum is presented. This is an observational study, and the amount of data available does not allow generalizing beyond some ‘common sense’ statements.

Most of the process areas of CMMI-DEV level 2 have been improved using Scrum. REQM, CM, and project management (PP and PMC) are strongly improved with Scrum. Others such as MA or PPQA are less covered.

Some issues were identified with respect to formal appraisals, and how to justify with Scrum that a specific goal of CMMI-DEV is implemented. Some Scrum practices did not leave a paper trail to demonstrate the implementation of some process areas of CMMI-DEV, and software companies may need to add artifacts to their Scrum practices to prepare for a formal appraisal.

In addition, the case study highlighted how the use of open-source tools was useful in improving process in the companies.

Based on the case study carried out, we can state that the quality of software processes may be improved using Scrum. And given these results, new case studies are planned for these organizations.

ACKNOWLEDGEMENTS

This research has been partially developed within the framework of the following projects: MODEL-CAOS (TIN2008-03582/TIN) financed by the Spanish Ministry of Education and Science.
REFERENCES


AUTHORS’ BIOGRAPHIES

Javier Garzás is PhD in Software Engineering (Cum Laude qualification), MS in Computer Science (graduated with honors and with first class standing), and Master in Enterprise Application Integration (awarded by PriceWaterhouseCoopers). Postdoctoral stay at Carnegie Mellon University (Pittsburgh, USA). Since 2006, he is CEO and Co-founder at Kybele Consulting (Software Engineering industry), spin-off of Rey Juan Carlos University, and since 2004, he is Associate Professor and Researcher at Rey Juan Carlos University. Lead Auditor of ISO 15504 SPICE and ISO 12207, Auditor of ISO 20000 – ITSMF, CISA, CGEIT and CRISC (ISACA), and CSQE (ASQ). Highlighting the following projects: participation in the application of software quality, outsourcing and software process improvement in more than 35 software companies. More than 60 research publications, 3 books edited, and more than 10 chapters written, and participation in more than 50 research projects of the European Union and Spanish Research Founding Agencies. Javier is author of www.javiergarzas.com, a blog about outsourcing, software quality and processes.

Dr. Mark C. Paulk is a Senior Systems Scientist at the Institute for Software Research in the School of Computer Science at Carnegie Mellon University. His research interests revolve around Total Quality Management, best practices (e.g., Software CMM, ISO/IEC 15504, and eSCM-SP), empirical research, high maturity practices, statistical thinking, and agile methods. He teaches in the Master of Software Engineering program at Carnegie Mellon. From 2002 to 2009, he worked at the IT Services Qualification Center at Carnegie Mellon. From 1987 to 2002, he worked at the Software Engineering Institute. He does part-time consulting and professional education via Paulk Consulting. Standard professional education offerings are listed on that site. He received his PhD in Industrial Engineering from the University of Pittsburgh, his MS in Computer Science from Vanderbilt University, and his BS in Mathematics and Computer Science from the University of Alabama in Huntsville. He is a Fellow of the ASQ, an ASQ Certified Software Quality Engineer, and a Senior Member of the IEEE.