# NDT-Agile: An Agile, CMMI-Compatible Framework for Web Engineering

Torrecilla Salinas, Carlos; Guardia, T.; De Troyer, Olga; Mejías, M.; Sedeño, J.

*Published in:*
Software Process Improvement and Capability Determination - 17th International Conference, SPICE 2017, Proceedings

*DOI:*
10.1007/978-3-319-67383-7_1

*Publication date:*
2017

*Document Version:*
Accepted author manuscript

# NDT-Agile: An Agile, CMMI-Compatible Framework for Web Engineering

C.J. Torrecilla-Salinas[1], T. Guardia[1], O. De Troyer[2], M. Mejías[1] and J. Sedeño[1]

[1] IWT2 Group, Universidad de Sevilla
{carlos.torrecilla, jorge.sedeno}@iwt2.org
risoto@us.es
[2]Department of Computer Science. Vrije Universiteit Brussel (VUB). Belgium
Olga.DeTroyer@vub.ac.be

**Abstract.** Agile and Web Engineering show important synergies, making Agile a common approach for Web development. Besides, several initiatives emerged to support CMMI-DEV within Agile, where CMMI-DEV aims to improve organizations' software development process. An approach integrating Agile, Web and CMMI-DEV might be of great value, since they might allow Web development teams to use Agile, as well as progress through CMMI-DEV maturity levels. For this purpose, we developed *NDT-Agile*, an NDT-based Agile framework to achieve the goals of CMMI-DEV in the context of Web Engineering. It was developed by mapping Agile practices to the goals of CMMI-DEV so as to identify existing gaps. Next, we searched for suitable Agile practices to cover the gaps and integrated them into a framework called *NDT-Agile*, which was validated using an expert-judgment technique: the Delphi method. This paper describes how we integrated Agile and CMMI-DEV into a Web Engineering framework. Besides, it also analyzes its initial evaluation, together with a first tool developed to support it.

**Keywords:** Agile, Scrum, CMMI, Web Engineering, Expert Judgment.

## 1 Introduction

Agile methodologies, i.e. those that can be grouped under the principles and values described in the Agile manifesto [1], emerged as an alternative to classic software development approaches, which were frequently based on heavy up-front planning and on freezing requirements before the development started. Agile brings a completely different view on how to handle and approach requirements [2]. It is based on improved communication, close collaboration with business representatives and reduced delivery cycles, among other elements [1]. Several approaches can be found within the label Agile, like Scrum [3], eXtreme Programming (XP) [4] or Kanban [5], being Scrum and XP the most popular ones [6, 7].

Web Engineering has established itself as the field of Software Engineering in charge of developing Web Systems, those conceived, developed, deployed and used on

the Web [8]. Several methodological approaches have been proposed for Web Engineering, such as Navigational Development Techniques (NDT) [9]. In turn, NDT proposes a Model-Driven approach to Web Engineering. NDT is a methodology that focuses on the first phases of the Web development lifecycle and utilizes a bottom-up process. It uses a highly-detailed requirement gathering phase guided by objectives with three sub-phases: requirements capturing, definition and validation. It is important to note that NDT was not developed bearing the Agile approach in mind.

Both Agile and Web Engineering emerged simultaneously and independently. However, they show great synergies [10]. As such, several Web development teams have already applied an Agile approach.

CMMI (Capability Maturity Model – Integration) [11] is a well-known approach designed to improve organizations' processes. Out of the different maturity models proposed by CMMI, CMMI-DEV (Capability Maturity Model – Integration for Development) is the one focusing on software development. Currently, CMMI is used in more than 5,000 companies [12]. The progress through the different CMMI maturity levels is normally associated with increases in quality and customer satisfaction [13].

During several years, the approaches proposed by Agile and by CMMI were seen as opposite and even contradictory [14], but after that initial reluctance period, initiatives emerged from both sides trying to find common grounds [15]. Recently, we can find several proposals trying to combine these approaches, both for generic software development projects [16] as well as for Web specific projects [17].

Summarizing all the aforementioned elements, we can conclude that an approach based on Agile principles that allows progressing through the different CMMI-DEV maturity levels, simultaneously supporting Web specificities, will be of great value as, on the one hand, it will enable organizations to keep using an Agile approach for their Web development projects but, on the other hand, it will ensure repeatability and institutionalization by means of the process improvement approach carried out by CMMI.

Based on the foregoing arguments, we have developed *NDT-Agile*, a framework conceived to help organizations achieve the specific and generic goals of CMMI-DEV in the context of Web Engineering, while keeping agility and ability to respond to changes. This paper has the following objectives: (i) present how agility, CMM-DEV and Web Engineering have been integrated into a coherent framework, called *NDT-Agile*; (ii) describe the assessment of *NDT-Agile* by means of an expert-judgment process based on the Delphi method [18]; (iii) introduce a first version of a tool to support *NDT-Agile;* and (iv) draw relevant conclusions and present further lines of research.

To achieve the listed objectives, the paper is organized as follows: after this introduction, Section 2 asks the research question and describes the utilized research approach. Section 3 discusses related work. Then, Section 4 introduces *NDT-Agile* by describing its main elements. Afterwards, Section 5 presents the expert-judgment validation process, as well as the developed supporting tool and, finally, Section 6 states the main conclusions and further lines of research.

## 2 Research question and research method

The main research question that we addressed in our research was: "*Can we develop an Agile approach compatible with CMMI-DEV and usable for organizations developing Web systems?*" To answer this question, we asked a few concrete research sub-questions:

- **RQ1:** *What are the existing gaps between the current most used Agile approaches and the specific and generic goals of CMMI-DEV for Web systems?*
- **RQ2:** *Are there any existing Agile techniques to cover those gaps, in case they exist?*
- **RQ3:** *How can we combine the characteristics of the most used Agile approaches with the Agile techniques identified in RQ2 in a single coherent framework suitable for Web systems development?*
- **RQ4:** *How can we validate this framework, in case it can be developed?*

Once the research questions were asked, the next step consisted in defining a suitable research approach, which finally comprised the following steps:

- **Perform a gap analysis:** To distinguish if the existing and most popular Agile approaches can cover the different specific and generic goals of CMMI-DEV maturity level. As previously stated, Scrum and XP are the most used Agile approaches [6] whose practices are mapped to CMMI-DEV. The results of our gap analysis will be presented in Section 3.
- **Identify suitable Agile techniques to cover the gap:** To identify goals not covered by Scrum or XP practices by means of the gap analysis. The next step consisted in searching (in existing Agile literature) for other suitable Agile practices to cover the gaps. Results of this exercise will be also presented in Section 3.
- **Combine the identified techniques in a single coherent framework:** To define a coherent framework, named *NDT-Agile*, the identified Agile practices should be combined in a suitable way, avoiding duplicities, gaps or contradictions. *NDT-Agile* will be presented in detail in Section 4.
- **Validate the proposed framework:** To validate the proposed framework by means of an expert judgment process based on the Delphi method [18], before performing real-life experiments. The goal of this expert evaluation was to obtain an initial validation of the framework. This process will be presented in Section 5.
- **Develop an initial version of a tool to support the framework:** To develop a first version of a supporting tool to backup future deployments of the framework. Such tool will be introduced in Section 5.

## 3 Related work

This section summarizes related work that was collected by means of a Systematic Literature Review (SLR) based on Kitchenham's approach [19]. This process is described in more detail in [17]. The main goal was to identify previous works tackling

the relations among Agile, Web and CMMI. From the results of the SLR, different types of papers were identified such as: other existing SLRs in the context [16]; papers tackling different angles of the research problem; and theoretical studies or case studies coming from both Agile [20, 21, 22] and CMMI side [23], some of them including the Web perspective [24, 25, 26], but some others not [20, 22]. In this section, we focus on those works that performed a gap analysis or a mapping between Agile and CMMI, regardless of whether or not they considered the Web perspective.

In [21], a Scrum-based model named Model C-S is presented. It maps the specific practices of CMMI levels 2 and 3 to Scrum ones. This model includes 123 practices, but excludes some CMMI-DEV process areas linked to organizational issues. The work comprises a mapping describing which practices are fully or partially covered, or they are not covered at all, and some ad-hoc modifications to Scrum. Besides, the proposed model incorporates supporting elements to deploy and assess the model together with two case studies.

Further on, [22] assesses, from a theoretical point of view, whether the standard Scrum practices can cover the goals of a set of CMMI-DEV process areas from maturity levels 2, 3 and 4 (those linked to project management). The work presents an analysis of the coverage provided by Scrum to 22 of CMMI-DEV practices, establishing whether they are fully or partially covered, or they are not covered at all.

In [20], a theoretical study on whether Scrum standard practices can cover the goals of a set of CMMI-DEV level 2 process areas is presented. It particularly analyzes Project Planning (PP), Project Monitoring and Control (PMC) and Requirements Management (REQM). It also includes a case study based on an internal project assessment.

In [27], the coverage provided by XP practices to CMMI levels 2 and 3 process areas is studied from a theoretical point of view. From its conclusions, we took out that XP supports most level 2 practices and some level 3 ones. The paper also highlights some limitations to CMMI coverage depending on the project size.

Additionally, [28] maps specific goals of CMMI maturity level 3 process areas to three different Agile methods (Scrum, XP and Kanban). Then, it evaluates which of the practices proposed by the analyzed Agile methods can cover the different goals of CMMI and provide a percentage of coverage to each of the analyzed techniques. The main conclusion of this work is that there is compatibility between Agile approaches and CMMI level 2, as many of the goals of maturity level can be covered. Finally, the paper includes a case study to validate the proposal.

After this review process, we concluded that all the above-described works focus on generic development and do not consider Web specificities. Moreover, we confirmed that all of them are partial, not presenting a full gap analysis or a complete mapping to all CMMI-DEV maturity levels' process areas.

As mentioned in Section 2, the first step in our research was conducting a gap analysis to identify whether the existing and most popular Agile approaches can cover the different specific and generic goals of CMMI-DEV maturity level. This analysis specifically included Web specificities. In [26], we compared Scrum practices with the goals of CMMI-DEV maturity level 2. This work analyzed theoretically the gap between Scrum practices and those of CMMI-DEV level 2, concluding that, even though there is no full coverage between both, they are highly compatible. In this paper, we

also included a proposal to extend Scrum with the aim to cover the identified gap. Later, in [25], we proposed a mapping between Agile practices (including Scrum and XP standard practices, but going beyond them) and goals of CMMI-DEV maturity level 3. The paper settles that Agile techniques and CMMI-DEV are still highly compatible. Finally, in [24] we performed a gap analysis between Scrum and XP, and the proposed goals of CMMI-DEV maturity levels 4 and 5. The main conclusion was that Scrum or XP standard practices do not cover CMMI practices. Based on that assumption, we identified a set of Agile practices that could be suitable to cover the gap. Table 1 summarizes the identified coverage (from [24, 25, 26]) to the different CMMI-DEV maturity levels. To obtain the percentage of coverage, the number of CMMI-DEV specific practices fully covered by Scrum/XP standard practices of a particular maturity level was calculated and then divided by the total number practices defined in the maturity level.

**Table 1.** Identified coverage of Scrum/XP per CMMI-DEV maturity level.

| CMMI-DEV level 2 | |
|---|---|
| Approach | Coverage |
| Scrum | 72.2% |
| XP | 66.7% |
| Combined Scrum/XP | 92.6% |

| CMMI-DEV level 3 | |
|---|---|
| Approach | Coverage |
| Scrum | 34.8% |
| XP | 54.7% |
| Combined Scrum/XP | 60.5% |

| CMMI-DEV level 4 | |
|---|---|
| Approach | Coverage |
| Scrum | 0.0% |
| XP | 0.0% |
| Combined Scrum/XP | 0.0% |

| CMMI-DEV level 5 | |
|---|---|
| Approach | Coverage |
| Scrum | 0.0% |
| XP | 0.0% |
| Combined Scrum/XP | 0.0% |

Table 2 describes the identified and proposed extensions spotted in [24, 25, 26] to complement Scrum and XP with the aim to cover all the goals of CMMI-DEV. It also indicates whether the proposal is either an existing Agile practice (and then points to a reference describing it), or it is just an ad-hoc modification:

**Table 2.** Identified proposed extensions to Scrum/XP standard practices.

| CMMI-DEV level 2 – Proposed extensions | |
|---|---|
| Extension | Type |
| Sprint 0 | Agile practice [26] |
| Establish measurement objectives, how to measure them and how to store measures and collect data during the project | Ad-hoc modification |
| Establish how, when and where to store the project data and use the selected sources during the project | Ad-hoc modification |
| Establish how to communicate and manage the project data and follow the agreed approach during the project | Ad-hoc modification |
| Establish quality objectives, briefly documenting the agreements | Ad-hoc modification |
| Agile contracts techniques | Agile practice [29] |

| CMMI-DEV level 3 – Proposed extensions | |
|---|---|
| Extension | Type |
| Agile Project Management | Agile practice [30] |
| Scrum at Enterprise Level | Agile practice [31] |
| Lean Software Development | Agile practice [32] |
| Agile Risk Management | Agile practice [33] |
| CMMI-DEV level 4 – Proposed extensions | |
| Extension | Type |
| Performance and KPI baselines | Ad-hoc modification |
| Adapt the process to achieve desired quality and performance objectives | Ad-hoc modification |
| Select measures and techniques to be used for quantitative management | Ad-hoc modification |
| Use Agile Performance Indicators | Agile practice [34] |
| Agile EVM | Agile practice [35] |
| CMMI-DEV level 5 – Proposed extensions | |
| Extension | Type |
| Lean Software Development | Agile practice [32] |

## 4 *NDT-Agile*: An Agile CMMI-compatible framework for Web Engineering

In the previous section, we identified works related to RQ1 and RQ2, including our own gap analysis and mapping exercise [24, 25, 26]. We also identified a suitable list of Agile practices or ad-hoc modifications that simultaneously support Web specificities and all the specific and generic goals of CMMI-DEV. Nevertheless, a list of practices is not useful for organizations that focus on looking for a coherent framework to implement and customize. In this section, we present *NDT-Agile*, an Agile framework built upon the conclusions of the gap-analysis and on top of Scrum and XP practices, including all identified proposed extensions listed in Table 2. *NDT-Agile* also supports Web specificities by integrating NDT (i.e. a Web development methodology) and incorporating it into an Agile lifecycle. The description looks at the way in which CMMI-DEV is supported. *NDT-Agile* is composed of three main components:

- ***NDT-Agile* lifecycle**, an iterative and incremental lifecycle that describes the way projects are identified, planned, approved and developed, and which encapsulates NDT techniques. It focuses on covering the lifecycle related to the goals of CMMI-DEV levels 2 and 3.
- **Agile complementary techniques**, based on techniques identified in the gap analysis and complementing the framework beyond the scope of a project lifecycle. They cover the remaining not organizational-related goals of CMMI-DEV.

- **NDT-Agile governance,** which wraps the previous two elements and ensures a proper framework rollout, customization and improvement. It covers those goals of CMMI-DEV levels 4 and 5 that have an organizational dimension.

Figure 1 represents graphically the elements of the framework. The different components are discussed in the next subsections.

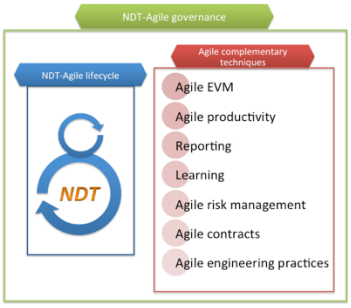


**Fig. 1.** *NDT-Agile* Components

### 4.1. *NDT-Agile* lifecycle

*NDT-Agile* lifecycle is used to manage projects (identify, plan and execute them) [35]. It comprises two main phases:

- **Project launching**, which is the only non-iterative phase of the framework, where an initial plan is developed by means of Agile estimation techniques [36] combined with Agile Project Management inception techniques [30]. It is presented in the form of an Agile project charter for the organization's management to approve.
- **Project development**, is an iterative phase that, based on the Scrum lifecycle, includes all identified modifications to achieve most project managements related to the goals of CMMI-DEV levels 2 and 3. After a ground-setting *Sprint 0*, a succession of Sprints is run with the aim to develop the project. The initial plan is adjusted Sprint-by-Sprint in order to ensure that business priorities are always identified and implemented. During this phase the requirements engineering, definition and validation proposed by NDT take place.

Fig. 2 depicts the proposed NDT-Agile lifecycle:

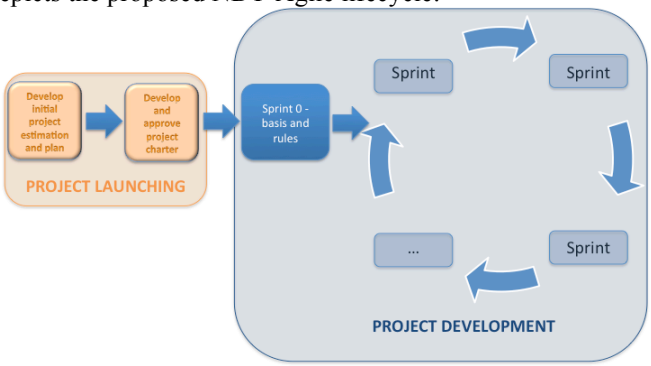


**Fig. 2.** *NDT-Agile* lifecycle [35]

### 4.2. Agile complementary techniques

Agile complementary techniques were incorporated into our framework aiming to cover all the goals of CMMI-DEV that go beyond the scope of a project lifecycle and mainly come from the identified list of Agile practices resulting from the gap analysis. We identified and included a total of seven complementary techniques:

- **Agile EVM** [35]: It provides an Agile way to control project constraints like budget and schedule without including extra overhead. They are included to cover the remaining goals of Quantitative Project Management (QPM) process area not covered by the standard Scrum/XP practices.
- **Agile productivity metrics** [34]: They are proposed to cover the goals of OPP (Organizational Project Performance), helping to measure consistently teams' productivity and ensure continuous improvement.
- **Agile reporting**: It is established to cover the generic goals of CMMI-DEV, as described in CMMI-DEV standard, and propose an Agile approach so as to enhance communication with the stakeholder. It includes classic Agile elements such as burn-down or burn-up charts.
- **Agile Learning**: Coming from the Lean approaches [32], it is proposed to fully cover the goals of Organizational Training (OT), Causal Analysis and Resolution (CAR) and Organizational Project Management (OPM). It contains techniques to ensure both team and organization improvements, by means of elements like retrospectives or communities of practices.
- **Agile risk management** [33]: It is included to meet the specific goals of RSKM (Risk Management) and provide Agile projects with explicit risk management capabilities without extra overhead.
- **Agile contracting** [29]. It is proposed to cover the goals of Supplier Agreement Management (SAM) and as a way to ensure an Agile relation with providers. That guarantees that risk is well balanced, thus all parties gain with such a relationship
- **Agile engineering practices** [4]: They mainly come from XP and are proposed to cover all CMMI-DEV engineering process areas, including Agile design, test and validation elements.

### 4.3. *NDT-Agile* governance

*NDT-Agile* governance is the third component of the framework. It is proposed to cover the goals of level 4 and 5 process areas. It is based on Schwaber's proposal to scale Scrum to organizational levels [31] and prescribe the establishment of a governance body (named Enterprise Transition Team), which is set at organization level. It comprises the following main objectives:

- Tailor the framework according to the organization's specific needs.
- Define the different organizational assets (tool or lessons learnt, among others).
- Establish project baselines and define organizational KPIs.

The proposed governance body also ensures that Agile practices like Scrum of Scrums (in order to coordinate the different existing Agile teams) or the maintenance of an organization wide product backlog (to have a view of the progress at organizational level) are established.

# 5 Validation of the approach and supporting tool

In this section, we explain how we carried out an initial validation of our proposal. For this purpose, an expert-judgment exercise based on the Delphi method was performed (Section 5.1). Furthermore, we also describe the first version of the supporting tool that enabled the framework deployment in practice (Section 5.2).

## 5.1. Expert-judgment process

As the implementation of new methods or frameworks within organizations always implies economical risks and presents organizational challenges, companies may be reluctant to incorporate them. If an initial validation conducted by a set of well-known experts is presented beforehand, some of these reluctances can be overcome and organizations might be more willing to experiment with the new working methods or frameworks. One of these expert-judgment techniques is the Delphi method [18], which consists in a panel of experts who, by means of structured and anonymous questionnaires and a series of rounds, reach consensus on a specific topic.

In order to validate *NDT-Agile* proposal using the Delphi method, a panel of 20 experts in one or more of the analyzed fields, coming from 8 different countries, was created and three consecutive rounds took place. The questionnaire used was composed of 21 statements, distributed among 4 different domains (i.e. dimensions) as follows:

- **Agile** dimension: 6 statements were used to assess the agility of the framework.
- **CMMI** dimension: 5 statements were used to evaluate the compliance with the different goals of CMMI maturity levels.
- **Web** dimension: 7 statements were provided to supervise the support given to Web specific characteristics.
- **Framework** dimension: 3 statements were used to test the internal coherence of completeness of *NDT-Agile*.

The questionnaire was made available to the experts in three rounds that were organized between February and June 2016. In each round, experts were asked to express their agreement with each of the statements by means of a Likert scale [37] ranging from “Complete disagreement” (value 1) to “Complete agreement” (value 5).

Two types of analysis were conducted in order to evaluate the results of the different rounds:

- **Descriptive analysis**, which assessed the level of agreement of the experts with the proposed statements and the internal grade of consensus reached by the panel by means of calculating the mean, median and standard deviation of

the grades given to each statement, as well as analyzing the experts' textual comments.

- **Homogeneity and concordance analysis**, which used statistical tools like Chronbach's alpha [38], Kendall's W [39] and Simple Correspondence Analysis [40], calculated by means of R [41], to check the degree of consensus and stability of the panel's opinion on the analyzed subject through the different rounds.

In order to interpret the obtained results, we defined *strong agreement* of the panel experts on one of the statements if: the mean of the given grades was above 3.7 (in a scale ranging from 0 to 5, being 5 the maximum value), the median was 4 (representing "Agreement") and at least 60% of the raters' score was 4 or 5 ("Agree" or "Strongly agree"), with a minimum of 12 experts providing an opinion. We also defined *slight agreement* on one of the statements if: the mean of the given grades was between 3.5 and 3.7, with a median equal or higher than 3.5 and at least 45% of raters' score was 4 or 5, with a minimum of 12 experts providing an opinion. Figure 3 displays the obtained results after the third round distributed by the defined dimensions:

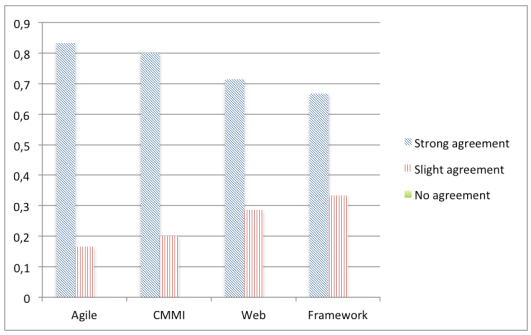


**Fig. 3.** Results by dimension

As Figure 3 shows, a high level of agreement was achieved for all the four analyzed dimensions, ranging from more than 80%, in the case of Agile, to more than 65%, in the case of the Framework dimension. It must be pointed out that no disagreement with the overall NDT-Agile proposal was identified from the selected panel. Table 3 also presents the overall results of the Delphi method after the third round:

**Table 3.** Delphi method: Overall results.

| Level of agreement | Number of statements | % |
|---|---|---|
| Strong agreement | 16 | 76.19% |
| Slight agreement | 5 | 23.81% |
| Agile contracts techniques | 0 | 0% |

Out of the 21 proposed statements, 75% of them showed strong agreement whereas 25% showed slight agreement. The main questions marks stated by the panel were

linked to feasibility of achieving the goals of CMMI-DEV level 5, support to Web systems' security and maintenance requirements within *NDT-Agile* and implementation of proposed governance model.

### 5.2. Supporting tool

Finally, and as an essential element to support deployment, a first version of a supporting tool was developed. For that purpose, we conducted a comparison exercise among existing Agile project tools available in the market, in order to find out the most suitable one that could fulfil our needs, instead of developing a completely new tool from the scratch. After assessing Mantis, JIRA, Bugzilla and Redmine [42], we chose the last one due to its active community, its plugin mechanism and our previous knowledge of the tool. Table 4 presents the results of our analysis, including the assessment criteria used:

- **License/Cost:** Is it an Open Source tool or does it offer a "free of cost/community" version?
- **Plugin schema:** Does the tool offer a plugin/extension mechanism?
- **Community:** Does the tool have a well-established community?
- **Agile:** Are there any Agile extensions available to be used?
- **Integration:** Is it possible to integrate the tool with other tools in an ecosystem?

**Table 4.** Tool assessment: Results.

| Tool | License/ Cost | Plugin | Community | Agile | Integration |
|---|---|---|---|---|---|
| Redmine | Open Source with Free/Community edition available | Yes | Yes | Yes | Yes |
| JIRA | Commercial tool | Yes | Yes | Yes | Yes |
| Bugzilla | Open Source with Free/Community edition available | No | Yes | Limited | No |
| Mantis | Open Source with Free/Community edition available | Yes | Yes | Yes | Yes |

After identifying a suitable tool, we defined a series of epics and user stories that allowed us to support our proposal and, among them, select the ones to be included in the first version of the tool. Basically, we chose those related to *NDT-Agile* lifecycle support and Agile EVM calculations. Once the scope of the first version was clearly identified, we tried to achieve the desired functionality by two different paths:

- Configuring and customizing Redmine, which let us cover a significant amount of functionality without further development.
- Developing a custom-made plugin, in order to achieve the remaining functionality.

## 6 Conclusions and future work

This paper presented *NDT-Agile*, an Agile, CMMI-compatible framework for Web Engineering. Its inception process was based on a complete gap analysis between

Scrum and XP and the different process areas of all CMMI-DEV maturity levels. The paper also provided an overview and justification of the initial validation of the framework, which was carried out by means of an expert-judgment process based on the Delphi technique. Finally, we briefly described the development of an initial version of a tool that could support the framework. In consequence, we were able to show that we could come up with *an integrated framework using an Agile approach, compatible with CMMI-DEV and usable for organizations developing Web systems*, providing an answer to our main research question. If we linked this work to the initially formulated research questions, we could state the following conclusions:

- *RQ1:* Several gaps were identified for all CMMI-DEV maturity levels. In the case of levels 2 and 3, we noticed that Scrum and XP are compatible with CMMI-DEV covering, either alone or combined, a significant amount of objectives. In the case of levels 4 and 5, we realized that there is no coverage at all, as those levels focus on organizational aspects, and Scrum and XP are more oriented towards operational ones.
- *RQ2*: A full list of complementary Agile techniques and modifications, suitable for Web systems, were identified to cover each of the gaps for all CMMI-DEV maturity levels.
- *RQ3*: We responded to this question by proposing *NDT-Agile*, a framework that, by means of an Agile lifecycle, ensures agility. It covers all remaining goals of CMMI-DEV, by including a set of complementary Agile techniques and a governance model, and supports Web specificities, by encapsulating NDT.
- *RQ4:* In order to perform an initial validation that afterwards would allow real-life deployments of the proposed framework, we conducted an expert judgment process based on the Delphi method. It offered promising results, as the identified panel agreed on the suitability of the approach. As a complementary element, we developed an initial version of a tool to encourage framework deployment.

As future lines of research we can highlight the improvement of the framework in those areas where experts expressed some concerns (such as security and maintenance practices, governance model or achievement of goals of CMMI-DEV level 5). Besides, the deployment of the framework in real-life projects and their assessment, via a formal SCAMPI process [43] or a self-assessment, remains to be done yet.

## Acknowledgements

This research has been supported by the MeGUS project (TIN2013-46928-C3-3-R), the Pololas project (TIN2016-76956-C3-2-R) and by the SoftPLM Network (TIN2015-71938-REDT) of the Spanish the Ministry of Economy and Competitiveness. We would also like to thank Dr. Diego Nieto Lugilde, Dr. Pedro Antonio García and Dr. Diego Torrecilla de Amo, from the University of Granada. Finally, we would like to show our gratitude to all experts participating in the Delphi method for their useful contributions.

## References


1. Beck, K. *et al*. “Manifesto for Agile Software Development”. 2001. From http://www.agilemanifesto.org. Last accessed 05-2017.
2. Schön, E. M., Thomaschewski, J., Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, *49*, 79-91.
3. Sutherland, J., Schwaber, K. “The Scrum Guide: The Definitive Guide to Scrum”. 2011. From http://www.scrum.org/Scrum-Guides. Last accessed 05-2017.
4. Beck, K. “Extreme Programming Explained: Embrace Change”. 2000. Boston: Addison-Wesley.
5. Anderson, D.J. “Kanban - Successful Evolutionary Change for your Technology Business”. 2010. Blue Hole Press.
6. Pikkarainen, M. *et al*. “The Impact of Agile Practices on Communication in Software Development”. Empirical Software Engineering, Springer, pp. 303-337. May. 2008.
7. VersionOne. “9th Annual State of Agile Survey”. 2015. From http://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf. Last accessed 05-2017.
8. Deshpande, Y.; Marugesan, S.; Ginige, A.; Hanse, S.; Schawabe, D.; Gaedke, M.; White, B. “Web Engineering”. Journal of Web Engineering. Vol. 1 No. 1. pp. 3-17.2002. Rinton Press.
9. Escalona, M.J.; Aragón, G. 2008. “NDT: A Model-Driven Approach for Web requirements”, IEEE Transactions on Software Engineering, 34(3). 2008. 370-390.
10. Mendes, E.; Mosley, N. “Web Cost Estimation: An Introduction”. Web Engineering: Principles and Techniques. IGI Global, pp 182-202. 2005.
11. CMMI Product Team. 2010. “CMMI for Development, Version 1.3.”. Nov. 2010. Carnegie Mellon University. Technical Report. From http://www.sei.cmu.edu/reports/10tr033.pdf. Last accessed 05-2017.
12. CMMI Institute. 2016. "Published Appraisal Results". From https://sas.cmmiinstitute.com/pars/. Last accessed 05-2017.
13. Goldenson, D. R. *et al*. "Why Make the Switch? Evidence about the Benefits of CMMI". From http://www.sei.cmu.edu/library/assets/evidence.pdf. Last accessed 05-2017.
14. Staples, M. *et al*. 2007. “An exploratory study of why organizations do not adopt CMMI”. J. Syst. Softw. 80, 6 (June 2007), 883-895.
15. Glazer, H. *et al.* 2008. “CMMI or Agile: Why Not Embrace Both!”. Nov. 2008. Carnegie Mellon University. From http://www.sei.cmu. edu/reports/08tn003.pdf. Last accessed 05-2017.
16. Selleri Silva, F. *et al*. 2015. “Using CMMI together with agile software development: A systematic review”. Information and Software Technology, Volume 58, February 2015, Pages 20-43.
17. Torrecilla Salinas, C.J *et al*. 2016. “Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review”. InfSoftwTechnol 71 (2016) 92–107.
18. Dalkey, N.C.; Helmer, O. “An experimental application of the Delphi method to the use of experts”, Manag. Sci. 9 (1963) 458–467.
19. Kitchenham, B. *et al.* 2009. S. “Systematic Literature Reviews in Software Engineering – A Systematic Literature Review”. Information and Software Technology 51 7–15. 2009.
20. Díaz, J.; Garbajosa, J.; Calvo-Manzano, J.A. 2009. “Mapping CMMI Level 2 to Scrum Practices: An Experience Report”. Jan. 2009. SPI 42 93-104.
21. Lukasiewicz, K.; Miler, J. 2012. "Improving Agility and Discipline of Software Development with the Scrum and CMMI," Software, IET, vol.6, no.5, pp.416, 422, October 2012.
22. Marcal, A.S.C. *et al*. 2008. “Blending Scrum Practices and CMMI Project Management Process Areas”. 2008. ISSE 4. 17-29.

23. Jakobsen, C. R.; Johnson, K. A. 2008. “Mature Agile With a Twist of CMMI”. Aug. 2008. In proceeding of Agile Conference 2008 (Toronto, Canada 04–08 Aug. 2008). AGILE ‘08. IEEE.
24. Torrecilla Salinas, C.J.; Sedeño, J.; Escalona, M.J.; Mejías, M. 2014. “An Agile approach to CMMI-DEV levels 4 and 5 in Web development projects”. In Information Systems Development (ISD2016 Proceedings). Katowice, Poland.
25. Torrecilla Salinas, C.J.; Sedeño, J.; Escalona, M.J.; Mejías, M. 2014. “Mapping Agile Practices to CMMI-DEV Level 3 in Web Development Environments”. In Information Systems Development: Transforming Organisations and Society through Information Systems (ISD2014 Proceedings). Varaždin, Croatia.
26. Torrecilla Salinas, C.J.; Escalona, M. J.; Mejías, M. 2012. “A Scrum-based Approach to CMMI Maturity Level 2 in Web Development Environments”. In proceeding of International Conference on Information Integration and Web-based Applications & Services 2012 (Bali, Indonesia December 3-5 2012). iiWAS, 12. ACM.
27. Paulk, M. C. 2001. “Extreme programming from a CMM perspective”. Software, IEEE, 18(6), 19-26.
28. Bougroun, Z. *et al*. 2014. “The projection of the specific practices of the third level of CMMI model in agile methods: Scrum, XP and Kanban”. In proceedings of Information Science and Technology (CIST), 2014 Third IEEE International Colloquium in (pp. 174-179). IEEE.
29. Medinilla, A. “Contratos ágiles”. http://www.slideshare.net/proyectalis/090603-contratos-giles. 2009. Last accessed 05-2017.
30. Highsmith, J. “Agile Project Management: Creating Innovative Products, Second Edition”. NJ: Addison-Wesley. 2009.
31. Schwaber, K. “The Enterprise and Scrum”. Redmond: Microsoft Press. 2007
32. Poppendieck, M.; Poppendieck, T. 2003. “Lean Software Development. An Agile Toolkit”. Boston: Addison-Wesley.
33. Cohn, M. “Managing Risk on Agile Projects with the Risk Burndown Chart”. From http://www.mountaingoatsoftware.com/blog/managing-risk-on-agile-projects-with-the-risk-burndown-chart. Last accessed 05-2017.
34. Downey, S.; Sutherland, J. “Scrummetrics for hyperproductive teams: how they fly like fighter aircraft”. In proceedings of the 45th Hawaii International Conference on System Science, 2012, Maui, Hawaii, USA, January 4–7 2012.
35. Torrecilla Salinas, C.J. *et al.* “Estimating, planning and managing Agile Web development projects under a value-based perspective”. InfSoftwTechnol 61 (2015) 124–144.
36. Cohn, M. “Agile Estimating and Planning”. NJ: Addison-Wesley. 2005.
37. Likert, R. 1932. “A technique for the measurement of attitudes”. Archives of psychology.
38. Cronbach, L. J. 1951. “Coefficient alpha and the internal structure of tests”. Psychometrika,16(3), 297-334.
39. Legendre, P. 2005. “Species associations: the Kendall coefficient of concordance revisited”. Journal of agricultural, biological, and environmental statistics, 10(2), 226-245.
40. Benzécri; J. P. "L’Analyse des Données. Volume II, L’Analyse des Correspondances". 1973.
41. R Core Team. 2016. “R: A language and environment for statistical computing”. R Foundation for Statistical Computing, Vienna, Austria. From https://www.R-project.org/, Last accessed 05-2017.
42. Redmine. 2016. From http://www.redmine.org. Last accessed 05-2017.
43. SCAMPI Upgrade Team. 2011. “Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A”. Carnegie Mellon University. From http://www.sei.cmu.edu/reports/11hb001.pdf. Last accessed 05-2017.