Agile-CMMI V2.0 Alignment: Bringing to Light the Agile Artifacts Pointed Out by CMMI

Valeria Henriquez*
(Universidad Politécnica de Madrid, Madrid, Spain
ORCID: 0000-0002-9003-2254
v.henriquez@alumnos.upm.es,
* Corresponding author)

Jose A. Calvo-Manzano
(Universidad Politécnica de Madrid, Madrid, Spain
ORCID: 0000-0002-2864-2203
joseantonio.calvomanzano@upm.es)

Ana M. Moreno
(Universidad Politécnica de Madrid, Madrid, Spain
ORCID: 0000-0001-8397-6794
ammoreno@fi.upm.es)

Tomas San Feliu
(Universidad Politécnica de Madrid, Madrid, Spain
ORCID: 0000-0002-6104-7430
tomas.sanfeliu@upm.es)

Abstract: Agile-CMMI alignment is becoming increasingly popular. The literature abounds with research and experience reports associating these two approaches. A new model, CMMI V2.0, and in particular CMMI-DEV V2.0, was published recently. This latest model only provides partial alignment guidelines with Agile, and the experiences reported in previous literature (referred to CMMI-DEV V1.3) are not directly applicable to CMMI-DEV V2.0. Therefore, agile organizations that want to align with CMMI-DEV 2.0, either from scratch or as a transition from CMMI-DEV 1.3, face a major challenge. In this paper, we apply the content analysis method to several CMMI Institute sources to bring to the light the Agile artifacts related to CMMI-DEV V2.0 Practices. As a result, we identify 31 Agile artifacts to address alignment with CMMI-DEV V2.0, passing from 0% to 41% coverage for Practices at Maturity Levels 2 and 3. These Agile artifacts can help to materialize the business value of CMMI-DEV V2.0 Practices and provide a starting point for agile organizations to define their roadmap. Consequently, the contribution of this paper is to present a detailed practice-level mapping based on which to design an Agile-CMMI V2.0 roadmap.

Keywords: Agile and CMMI-DEV V2.0, CMMI V2.0, Agile Software Development, Software Process Improvement.

1 Introduction

Since the publication of the Agile Manifesto in 2001 (https://agilemanifesto.org/), Agile approaches have become more and more popular [1][2]. According to the 15th Annual State of Agile Report [3], Agile is nowadays mainstream. Of the surveyed organizations, 94% work with agile frameworks, mainly Scrum or Scrum-based methods like Scrumban and Scrum and XP [3].

On the other hand, CMMI (Capability Maturity Model Integration) is one of the most adopted performance improvement models worldwide, and its use has grown steadily over recent years [4]. In particular, CMMI-DEV, which focuses on providing best practices for software development, is the model most commonly adopted by organizations [5]. As a result, there are many reports of CMMI adaptation to fit different contexts, such as SMEs [6], risk management [7] or other models [8]. Nowadays, for instance, CMMI is predominant in Agile settings, and over 80% of CMMI-DEV appraisals in 2019 were at agile organizations [4].

Although CMMI adoption offers organizations attractive incentives, such as increased productivity, on-time delivery, or defect reduction [9], its adoption in Agile contexts has been an arduous task [10]. Organizations have to make a substantial effort to implement these hybrid processes effectively [10][11]. However, agile organizations are willing to put in this effort in order to: 1) introduce structure and

predictability, along with flexibility and agility, 2) scale practices, and 3) gain recognition from the market and differentiation from their competitors [12].

In this context, the publication of the latest version of CMMI, CMMI V2.0, in 2018 [13] posed the challenge to agile organizations of how to align with this model either from scratch or as a transition from CMMI-DEV V1.3.

The CMMI Institute has claimed to include specific guidelines for helping organizations using Scrum-based Agile artifacts to improve performance [14]. In this sense, CMMI-DEV V2.0 provides recommendations about how most of its Practice Areas (PAs) can relate to one or more Scrum-based Agile artifacts (PAs have context-specific sections containing the agile guidance, "Agile with Scrum Guidance", referring to some Agile artifacts). However, organizations evolve by adopting Capability Level Practices, and CMMI does not offer information about Agile artifacts related to CMMI-DEV V2.0 Practices [15]. This research aims to fill this gap, as shown in Figure 1.

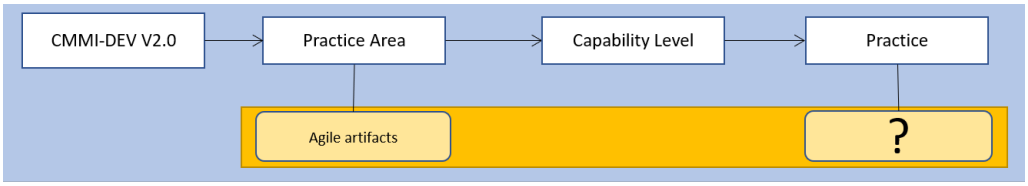

Figure 1. Problem to be solved.

One approach to address this gap would be to use the knowledge provided in the recent literature on Agile-CMMI alignment. However, as we will see later, existing knowledge is neither generally applicable, nor based on the latest version of CMMI, CMMI-DEV V2.0. Systematic literature reviews regarding Agile-CMMI alignment [16–20] highlight that the available literature is either theoretical, without further validation, or based on a few case studies of ad hoc implementations. As a result, each paper shows a different mapping between Agile artifacts (either practices, techniques, or assets) and CMMI-DEV V1.3 Specific Practices and suggests particular extensions to increase coverage. Even if an organization does decide to implement these extensions, results applicable to CMMI-DEV V1.3 are not directly applicable to CMMI-DEV V2.0 due to the structural differences between the two versions [21].

On the other hand, there are, to date, hardly any published studies applying CMMI V2.0 in Agile settings. As discussed later, there is some incipient research on this topic. At the time of writing, however, we had found only one journal publication reporting the experience of adopting CMMI V2.0 in an Agile context [22]. This experience report does not provide details about specific Agile artifacts to be used for the different CMMI V2.0 Practices.

In sum, existing knowledge only addresses the Agile-CMMI-DEV V2.0 alignment challenge partially (at PA level), leaving out the identification of the Agile artifacts related to the CMMI-DEV V2.0 Practices that an organization might need to implement. Therefore, it would be quite beneficial to have a toolkit of Agile artifacts identified at the practice level. This is the main contribution of this paper.

To address this issue, we apply the content analysis method [23] to several CMMI Institute sources. Our goal is to use this method to bring to light the Agile artifacts pointed out by the CMMI Institute and associate a set of these artifacts with CMMI-DEV V2.0 Practices. Notice that content analysis is a method that has been used in related investigations for different purposes, for example, to compare the structure of different maturity models [24][25] or to tailor CMMI Practices to quality models [26]. In our case, we use content analysis to identify and quantify Agile artifacts associated with CMMI-DEV V2.0 Practices.

These results would benefit agile organizations by:

(i) Facilitating the transition from CMMI-DEV V1.3 to CMMI-DEV V2.0, as it will be possible to identify which Agile artifacts used in CMMI-DEV V1.3 are associated with the new CMMI-DEV V2.0 model at the Practice level.

(ii) Providing a starting point for the adoption of CMMI-DEV V2.0, as it will be possible to identify a preliminary set of Agile artifacts associated with CMMI-DEV V2.0 Practices. This preliminary set of Agile artifacts will need to be tailored to each organization's business needs according to the PAs to be addressed.

The preliminary set of Agile artifacts associated with CMMI-DEV V2.0 Practices identified by this research constitutes neither a two-way relationship nor a prescription. Furthermore, CMMI operates on the *what* layer and Agile, on the *how* layer. Therefore, the Agile artifacts discussed in this paper are recommendations rather than instructions for achieving a particular CMMI maturity level. Our research aims to specify the content of the guidelines provided by the CMMI Institute in terms of agile recommendations. Therefore, this paper facilitates the definition of a roadmap for adopting CMMI-DEV

V2.0 in agile organizations. Each roadmap should be specified and motivated by the business needs of each organization and promoted by a change in the underlying organizational culture to embrace agility and a continuous improvement mindset.

The remainder of the paper is organized as follows. Section 2 discusses the related work on Agile-CMMI alignment, Section 3 describes the process used to achieve our aim, Sections 4, 5, 6, and 7 describe each of the steps of this process and their results, Section 8 discusses the practical implications of the results and sets out some points to be considered concerning its use in organizations. Finally, Section 9 outlines the conclusions and future work.

2 Related Work

As CMMI was originally sponsored by the United States Department of Defense (DoD) and quickly adopted by large companies using traditional software development methodologies, CMMI and Agile were regarded as opposite approaches [12].

While CMMI and Agile artifacts focus on different layers (what to do and how to do it, respectively), they are complementary, as they both provide values, principles, and best practices for achieving organizational business goals [27]. Therefore, there is now agreement on the fact that the alignment of the two approaches is possible and beneficial [28]. The growth in CMMI-DEV certifications at agile organizations over the last decade [29] is further evidence: fewer than 30% of CMMI-DEV-certified organizations employed Agile practices in 2009, whereas agile organizations accounted for 70% of CMMI-DEV certifications by 2016 and over 80% by 2018 [29].

In the related literature, there are many research papers and experience reports regarding the alignment between CMMI and Agile approaches (mainly CMMI-DEV V1.3 with Scrum and XP, respectively). We have found two recent literature reviews addressing this issue [30][15]. The first review [30] provides a more descriptive analysis of the literature, mentioning the main challenges. The second review [15], focuses on challenges but also on how CMMI V2.0 contributes to resolving the challenges discussed in the literature. For instance, Henriquez et al. [15] found five systematic literature reviews (SLR) analyzing CMMI-Agile alignment and covering 139 unique papers published between 2010 and 2016 [16–20]. These SLRs and other recent related papers pinpoint challenges at the tactical and organizational levels [15].

From the tactical perspective, the SLRs highlight a lack of prescriptiveness, where papers tend to provide limited recommendations about different agile practices or tools to be used [16][17][18][20]. Additionally, the alignment recommendations include associations between different Agile artifacts and CMMI-DEV V1.3 components in diverse formats. Therefore, the recommendations are not comparable [18]. In terms of coverage, many researchers identified gaps between CMMI-DEV V1.3 Specific Practices and Scrum and XP events, techniques, and assets. They conclude that these Agile artifacts are well matched at project level, but CMMI-DEV V1.3 Specific Practices at the top of the maturity level hierarchy remain uncovered [16][18].

From the organizational perspective, Agile-CMMI alignment is a difficult undertaking that requires a cultural change. This means that organizations are capable of adopting the successful experiences derived from Agile-CMMI projects [15]. In particular, papers detail the complexities related to tailoring guidelines to address the range of projects executed by the organization [17] [19]. Besides, they pinpoint the importance of systematically disseminating Agile and CMMI alignment experiences within senior management to garner (often lacking) support and involvement [15]. Some papers also highlight the risk of losses in terms of agility due to such alignment [17][18][19].

The literature over the past five years also contains individual studies regarding the application of CMMI-DEV V1.3 with Scrum, and XP. These studies again mention the above challenges [15]. For instance, Carlson and Soukup [11] present an ad hoc implementation of CMMI-DEV V1.3 with Scrum at Maturity Level 3, and Bahaa et al. [31] show the coverage that agile practices provide for Project Management Specific Practices in CMMI-DEV V1.3. Even so, the recent literature suggests an improvement in terms of papers tackling high CMMI-DEV V1.3 maturity levels and thus improving coverage. Sharma et al. [28] report a successful case study of combining Scrum and CMMI-DEV V1.3 at Maturity Level 5. This case study shows how high levels of CMMI can work together with Scrum, and how the data gathered in each iteration provide insight into processes that support the achievement of business objectives [28]. After performing a gap analysis, Torrecilla-Salinas et al. [32] concluded that neither Scrum nor XP cover any of the Maturity Level 4 and 5 Specific Practices. Then, they proposed Lean-Agile techniques and ad hoc extensions to achieve CMMI-DEV V1.3 Maturity Levels 4 and 5 in Web environments [32]. These proposals were not evaluated on CMMI appraisals but were validated by

experts [32] [33]. In the same vein, Kawamoto and De Almeida proposed Scrum-DR, a Scrum-based extension, formed by phases, roles, artifacts, and events to fill the gaps between Scrum and CMMI-DEV V1.3 without losing agility [34]. Additionally, Albuquerque et al. [35] conducted four case studies investigating the adoption of Agile and CMMI-DEV V1.3 beyond CMMI appraisals. They concluded that Agile-CMMI alignment is difficult to maintain and does not last long after appraisal [35].

Complementary to the above literature, the CMMI Institute published *A Guide to Scrum and CMMI: Improving Agile Performance with CMMI* in 2016 [12]. This guide has not been widely applied. A search of its title on Google Scholar returns only six search results, none of which report experiences of its use.

Later, CMMI V2.0 was published in 2018, but there is hardly any literature applying its guidelines in Agile contexts. A Google Scholar search for the terms "Agile" and "CMMI V2.0" retrieved 35 papers. However, only one [22] is a journal paper that describes an alignment experience of CMMI V2.0 and Agile, in particular, the Scaled Agile Framework (SAFe). This paper offers an overview of Agile-CMMI V2.0 alignment and does not provide sufficient information about how to nail down each CMMI V.20 Practice with SAFe artifacts. In the same vein, the CMMI Institute Resource Center has published short reports outlining successful experiences of Agile-CMMI V2.0 alignment in real settings, such as the experience of a digital products company in improving the predictability of its Agile process with CMMI V2.0 [36]. Again, these reports do not provide detailed information of how they achieve the Agile-CMMI alignment.

In sum, although there is a vast body of literature on Agile-CMMI alignment, the new CMMI V2.0, and in particular CMMI-DEV V2.0, remains unaddressed. Most of the available literature in this regard is outdated (does not focus on the latest CMMI version), and contributions to the field mainly focus on theoretical proposals and ad hoc implementations whose outcomes are not comparable. Therefore, agile organizations adopting CMMI-DEV V2.0 do not have access to recommendations about which Agile artifacts are possibly related to the particular CMMI-DEV V2.0 Practices they need to address.

3 Research Method

In order to contribute to the Agile-CMMI V2.0 alignment challenge, we apply content analysis [23] to the documents provided by the CMMI Institute. Content analysis is commonly used to analyze models and standards [24][25]. Figure 2 illustrates the research method followed. This figure shows the CMMI source documents used (top) and the steps taken (bottom).

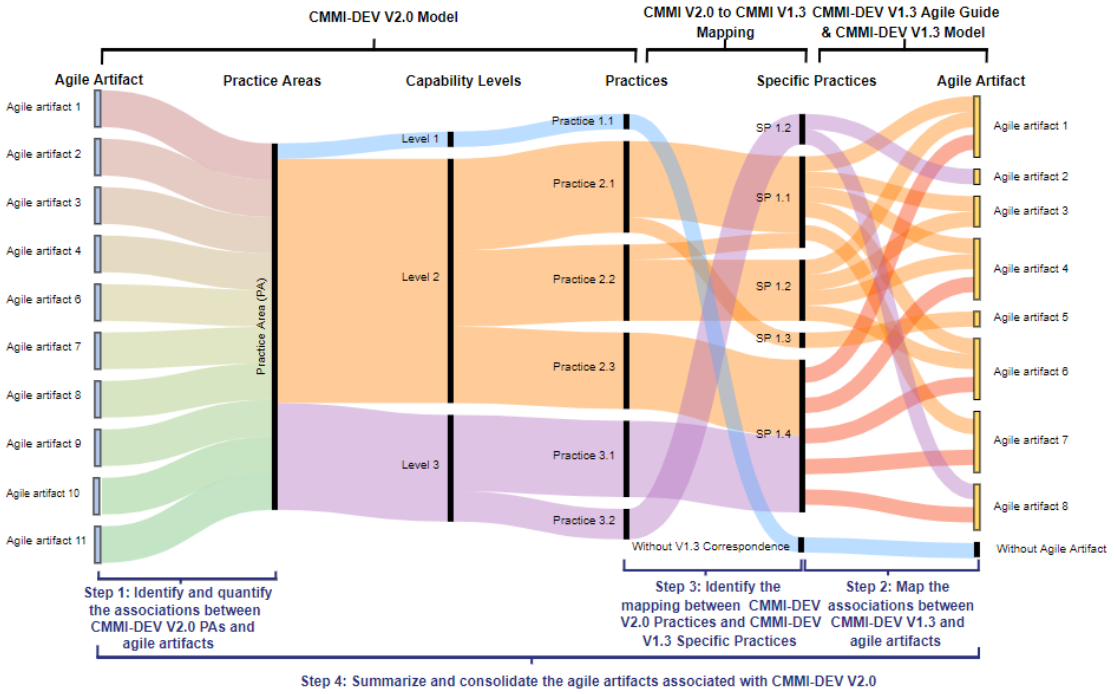

Figure 2. Research method.

- **Step 1:** Identify and quantify the associations between the CMMI-DEV V2.0 PAs and Agile artifacts by content analysis of the context-specific sections for agile contexts provided as part of the CMMI V2.0 model [37].
- **Step 2:** Identify and quantify the associations between the CMMI-DEV V1.3 Specific Practices and Agile artifacts. To do this, we use the results of the content analysis of the *A Guide to Scrum and CMMI: Improving Agile Performance with CMMI* document [12] (which maps 20 Agile artifacts to one or more CMMI-DEV V1.3 Specific Practices) in conjunction with the CMMI-DEV V1.3 model [38].
- **Step 3:** Identify and quantify the CMMI-DEV V2.0 Practices associated with CMMI-DEV V1.3 Specific Practices. To do this, we use the *CMMI V2.0 to V1.3 Detailed Practice Mapping* document [39].
- **Step 4.** Summarize and consolidate the Agile artifacts associated with CMMI-DEV V2.0 at PA level and at Practice level. To do this, we use the results of Steps 1, 2, and 3.

Following Denscombe's recommendations [23], content analysis can be divided into six activities. Table 1 summarizes the decision made as part of each of the content analysis activities applied within this study. Each table column shows an analysis activity, and each row shows how these activities were applied to the different documents that we used for the analysis.

Choose an appropriate sample of text	**Break down into small components**	**Develop relevant categories for data analysis**	**Code the units in line with the categories**	**Count the frequency with which these units occur**	**Analyze the text in terms of the frequency of the units and their relationships**
CMMI V2.0 model [37]	Each context-specific section labelled as *Agile with Scrum Guidance*	Agile artifacts (practices, techniques, and assets).	Code each mentioned agile artifact	Consolidate the mentioned Agile artifacts and count the number of CMMI V2.0 components in which they are mentioned	Identify and count the CMMI-DEV V2.0 components associated with agile artifacts (see Section 4 for detailed results).
A Guide to Scrum and CMMI: Improving Agile Performance with CMMI [12]	Each table in the Integrating CMMI with Agile Ceremonies & Techniques section	CMMI V1.3 Specific Practices	Code each mentioned Specific Practice	Count the Specific Practices and count the number of agile artifacts in which they are mentioned	Identify and count the CMMI-DEV V1.3 components associated with agile artifacts (see Section 4 for detailed results).
CMMI for Development, Version 1.3 [38]	Each section labelled as Specific Practices by Specific Goal (SG)	CMMI V1.3 Specific Practices	Code each mentioned Specific Practice	Count all Specific Practices mentioned in CMMI-DEV V1.3	This analysis complements the previous analysis to improve the understanding of the number of Agile guidelines in terms of the whole CMMI-DEV V1.3 model.
CMMI V2.0 to V1.3 Detailed Practice Mapping [39]	The content of each row related to a CMMI-DEV V2.0 Practice	CMMI-DEV V1.3 Specific Practices	Code each mentioned Specific Practice	Count all CMMI-DEV V2.0 Practices related to CMMI-DEV V1.3 Specific Practices	Identify and count the CMMI-DEV V2.0 Practices associated with CMMI-DEV V1.3 Specific Practices. Used in conjunction with the previous analysis, this result identifies which CMMI-DEV V2.0 Practices have the potential to be enriched with the Agile artifacts associated with their Specific Practices.

Table 1. Content analysis activities and documents.

Below we describe in detail the different steps shown in Figure 2 along with their respective results.

4 Identifying the Associations between CMMI-DEV V2.0 and Agile Artifacts

This section describes Step 1 in Figure 2. To identify the associations between CMMI-DEV V2.0 and Agile artifacts, we analyzed the context-specific sections of the CMMI-DEV V2.0 model [37] that include guidance for Agile settings. In particular, we analyzed the subsections labelled by the CMMI Institute as *Agile with Scrum Guidance*. These subsections are to be found only in the CMMI-DEV V2.0 PAs and are not available at the Practice level. Consequently, the 186 Practices making up the CMMI-DEV V2.0 model PAs are not explicitly associated with agile practices.

Table 2 shows the set of 30 Agile artifacts identified for CMMI-DEV V2.0 PAs. The blue cells show the association of an agile artifact (column) with a PA (row). Note, for example, Release Planning is associated with Technical Solution (TS) and another 10 PAs. Most of the CMMI-DEV V2.0 PAs are associated with Agile artifacts. However, there are also some cases where the PAs are associated with work products derived from the execution of an agile practice. For example, the Causal Analysis and Resolution (CAR) PA is associated with the data derived from holding Retrospective Meetings rather than with the activity itself. In these cases, the work product has been considered as an agile artifact (and the Retrospective Data artifact is, therefore, associated with the Causal Analysis and Resolution PA in Table 2).

Additionally, Table 2 shows that 16 out of the 19 CMMI-DEV V2.0 PAs are associated with one or more Agile artifacts. In particular, the three PAs that are not associated with Agile artifacts are Organizational Training (OT), Governance (GOV), and Implementation Infrastructure (II).

In CMMI-DEV V2.0, the Governance (GOV), and Implementation Infrastructure (II) are crucial, as they are required to achieve any maturity or capability level [37]. The CMMI Institute stipulates that these two PAs replace CMMI-DEV V1.3 Generic Goals and Generic Practices [14].

As CMMI-DEV V2.0 Agile guidelines are provided at PA level, it might be argued that this guidance also covers Practices. However, CMMI V2.0 does provide practice-level contextualizations for contexts other than Agile with Scrum Guidance, such as Services. This suggests that associations between Agile artifacts and CMMI-DEV 2.0 Practices would be helpful.

	List of Agile artifacts identified for CMMI-DEV V2.0																													
Practices Area of CMMI-DEV V2.0	**Release Planning**	**Sprint Planning**	**Sprint Execution**	**Sprint Review/Demo**	**Retrospective**	**Daily Scrum**	**Backlog Grooming**	**Scrum of Scrums**	**DevOps**	**User Stories**	**Epics**	**Definition of Done**	**Code Review**	**Continuous Integration /Build**	**Team Estimation Game, planning poker**	**Pair Programming**	**Refactoring**	**Team Agreements**	**Technical Debt**	**Historical Velocity**	**Agile Process Definition**	**Product Backlog**	**Sprint Backlog**	**Release Plan**	**Task Board**	**Burndown Charts**	**Source Code & Executables**	**Acceptance Criteria/Test**	**Test Plan, Cases & Result**	**Retrospective Data**
Technical Solution (TS)	■	■	■	■	■		■																							
Product Integration (PI)									■					■																
Requirements Development and Management (RDM)	■	■	■	■	■		■			■	■	■			■							■	■					■	■	
Peer Reviews (PR)	■	■	■	■			■									■														
Verification and Validation (VV)			■	■								■										■	■					■	■	
Process Quality Assurance (PQA)	■	■	■	■	■	■	■						■	■	■															■
Monitor and Control (MC)			■	■	■	■		■														■	■		■	■				
Planning (PLAN)	■	■	■	■			■											■				■	■		■					
Estimating (EST)	■	■					■			■	■				■					■		■	■							
Causal Analysis and Resolution (CAR)			■																											■
Configuration Management (CM)	■	■	■	■	■		■					■									■	■	■	■	■	■	■		■	■
Decision Analysis and Resolution (DAR)	■	■	■				■															■	■	■			■			
Process Asset Development (PAD)	■	■	■	■	■		■										■				■	■	■	■	■					■
Managing Performance and Measurement (MPM)	■	■	■	■	■	■	■										■			■						■				
Process Management (PCM)		■	■	■	■																									■
Risk and opportunity Management (RSK)	■	■	■	■	■	■	■										■		■							■				
Organizational Training (OT)																														
Governance (GOV)																														
Implementation Infrastructure (II)																														

Table 2. Association between CMMI-DEV V2.0 Practice Areas and Agile artifacts pointed out by the CMMI Institute [37].

5 Identifying the Associations between CMMI-DEV V1.3 and Agile Artifacts

This section describes Step 2 of Figure 2. To do this, we analyzed the document entitled *Guide to Scrum and CMMI: Improving Agile Performance with CMMI* [12] to identify and quantify the associations between Agile artifacts and CMMI-DEV V1.3 Specific Practices.

This guide contains recommendations for aligning 20 Agile artifacts with CMMI-DEV V1.3 Generic Goals, Generic Practices, and Specific Practices. However, most of these recommendations refer to associations between Agile artifacts and CMMI-DEV V1.3 Specific Practices. Therefore, we focus on the associations between these components.

Table 3 illustrates one example of the relationship between the Backlog Grooming and the Requirements Management (REQM), Requirements Development (RD), and Project Planning (PP) Specific Practices, as specified in [12].

Agile Practice	Process Area	Specific Practice
Backlog Grooming	Requirements Management (REQM)	Understand Requirements
		Obtain Commitment to Requirements
		Manage Requirements Changes
		Maintain Bi-Directional Traceability
		Ensure Alignment Between Project Work & Requirements
	Requirements Development (RD)	Elicit Needs
		Transform Stakeholder Needs into Customer Requirements
		Establish Product and Product Component Requirements
		Identify Interface Requirements
		Establish Operational Concepts and Scenarios
		Establish a Definition of Required Functionality and Quality Attributes
		Analyze Requirements
		Analyze Requirements to Achieve Balance
		Validate Requirements
	Project Planning (PP)	Estimate the Scope of the Project
		Establish Estimates of Work Product and Task Attributes
		Estimate Effort and Cost
		Control Work Products

Table 3. Associations between the Backlog Grooming Agile practice and CMMI-DEV V1.3 Specific Practices. Information excerpted from [12].

The *Guide to Scrum and CMMI: Improving Agile Performance with CMMI* document [12] is structured to identify Agile artifacts associated with CMMI-DEV V1.3 Specific Practices. On this ground, the result of the content analysis of [12]needs to be scrutinized in the opposite direction using the CMMI-DEV V1.3 model [38] in order to determine if there are associations for all CMMI-DEV 1.3 Specific Practices and whether these associations apply to more than one Agile artifact. In other words, we identified the Agile artifacts associated with each of the 167 Specific Practices of the CMMI-DEV V1.3 model, if any. Due to the size of the model (22 Process Areas and 167 Specific Practices), the detailed result identifying all the Specific Practices and their associations with Agile artifacts is available at https://short.upm.es/59y2y.

Table 4 shows an example, summarizing the associations between Agile artifacts identified in the guidance for all CMMI-DEV V1.3 Project Planning (PP) Process Area Specific Practices (14 SPs). The yellow cells in Table 4 highlight the association between each Specific Practice (rows) and one or more Agile artifacts. For example, the Backlog Grooming artifact is associated with Estimate the Scope of the Project (PP 1.1), Establish Estimates of Work Product and Task Attributes (PP 1.2), and Estimate Effort and Cost (PP 1.4) Specific Practices. Additionally, we find that 11 out of the 14 Project Planning (PP) Specific Practices are associated with Agile artifacts.

		Agile Artifacts								
Process Area	**Specific Practices**	Backlog Grooming	Definition of Done	Product Backlog	Release Planning	Sprint Planning	Sprint / Iteration	Team Agreements	Team Estimating Game & Planning Poker	Technical Debt
Project Planning (PP)	Estimate the Scope of the Project (PP 1.1)									
	Establish Estimates of Work Product and Task Attributes (PP 1.2)									
	Define Project Lifecycle Phases (PP1.3)									
	Estimate Effort and Cost (PP 1.4)									
	Establish the Budget and Schedule (PP 2.1)									
	Identify Project Risks (PP 2.2)									
	Plan Data Management (PP 2.3)									
	Plan the Project's Resources (PP 2.4)									
	Plan Needed Knowledge and Skills (PP 2.5)									
	Plan Stakeholder Involvement (PP 2.6)									
	Establish the Project Plan (PP 2.7)									
	Review Plans That Affect the Project (PP 3.1)									
	Reconcile Work and Resource Levels (PP 3.2)									
	Obtain Plan Commitment (PP 3.3)									

Table 4. Association between all the CMMI-DEV V1.3 Process Planning (PP) Specific Practices and Agile artifacts pointed out by the CMMI Institute [12].

The quantitative result of the associations between the above Agile artifacts and the entire CMMI-DEV V1.3 model is shown in Table 5. The first column in Table 5 shows the CMMI-DEV V1.3 Process Area. The second column shows the proportion of Specific Practices associated with Agile artifacts over the total number of Specific Practices for each Process Area. Cells representing at least one association between Specific Practices and Agile artifacts are highlighted in yellow. Following CMMI Institute notation, the Process Areas and Specific Practices of CMMI-DEV V1.3 Maturity Levels 2, 3, 4, and 5 are outlined in Table 5 by orange, violet, red and green lines, respectively. The bottom row in Table 5 denotes the percentage of Specific Practices associated with Agile artifacts in each of the above maturity levels. Note that each maturity level includes all lower maturity levels. We find, for example, that of the 140 Specific Practices at Maturity Level 3 (54 SPs from Maturity Level 2, plus 86 SPs from Maturity Level 3), 73 (30 SPs from Maturity Level 2, plus 43 from Maturity Level 3) are associated with Agile artifacts, resulting in a rate of association of 52%. Additionally, we observe that no associations were identified between Agile artifacts and Specific Practices proper to Maturity Levels 4 and 5. Therefore, the number of associations for the above maturity levels are unchanged at 73 Specific Practices and match up with the associations identified for the lower maturity level (Maturity Level 3). Notice that "no associations" means that the analyzed documents do not provide information on how to materialize the Specific Practices of CMMI Maturity Levels 4 and 5.

CMMI-DEV V1.3 Process Areas	Proportion of Specific Practices associated with Agile artifacts over the total number of Specific Practices in the Process Area			
Causal Analysis and Resolution (CAR)				0/5
Quantitative Project Management (QPM)				0/7
Organizational Process Performance (OPP)			0/5	
Organizational Performance Management (OPM)			0/10	
Decision Analysis and Resolution (DAR)		0/6		
Integrated Project Management (IPM)		9/10		
Organizational Process Definition (OPD)		2/7		
Organizational Process Focus (OPF)		0/9		
Organizational Training (OT)		0/7		
Validation (VAL)		5/5		
Technical Solution (TS)		2/8		
Risk Management (RSKM)		3/7		
Product Integration (PI)		5/9		
Requirements Development (RD)		10/10		
Verification (VER)		7/8		
Configuration Management (CM)	0/7			
Project Planning (PP)	11/14			
Measurement and Analysis (MA	7/8			
Project Monitoring and Control (PMC)	7/10			
Process and Product Quality Assurance (PPQA)	0/4			
Requirements Management (REQM)	5/5			
Supplier Agreement Management (SAM)	0/6			
	ML2 55% - (30/54) SP	ML3 52% - (73/140) SP	ML4 47% - (73/155) SP	ML5 44% - (73/167) SP

Table 5. Quantitative result of associations between CMMI-DEV V1.3 Specific Practices and Agile artifacts.

6 Identify the CMMI-DEV V2.0 Practices Associated with CMMI-DEV V1.3 Specific Practices

This section describes Step 3 of Figure 2. To map CMMI-DEV V2.0 Practices to the Agile artifacts identified for CMMI-DEV V1.3, we need to have a grasp of the mapping between the two models. To do this, we work with the *CMMI V2.0 to V1.3 Detailed Practice Mapping* document [39].

This mapping relates CMMI V2.0 Practices to Specific Practices of version 1.3 of CMMI-DEV, CMMI-SVC, or CMMI-ACQ. As this paper focuses on software development, we have only considered CMMI-DEV V2.0 Practices related to CMMI-DEV V1.3 Specific Practices.

Figure 3 illustrates the result of the analysis of the information provided by the above mapping (Step 3). This diagram shows an example of how the CMMI-DEV V2.0 Estimating (EST) Practices map to CMMI-DEV V1.3 Specific Practices.

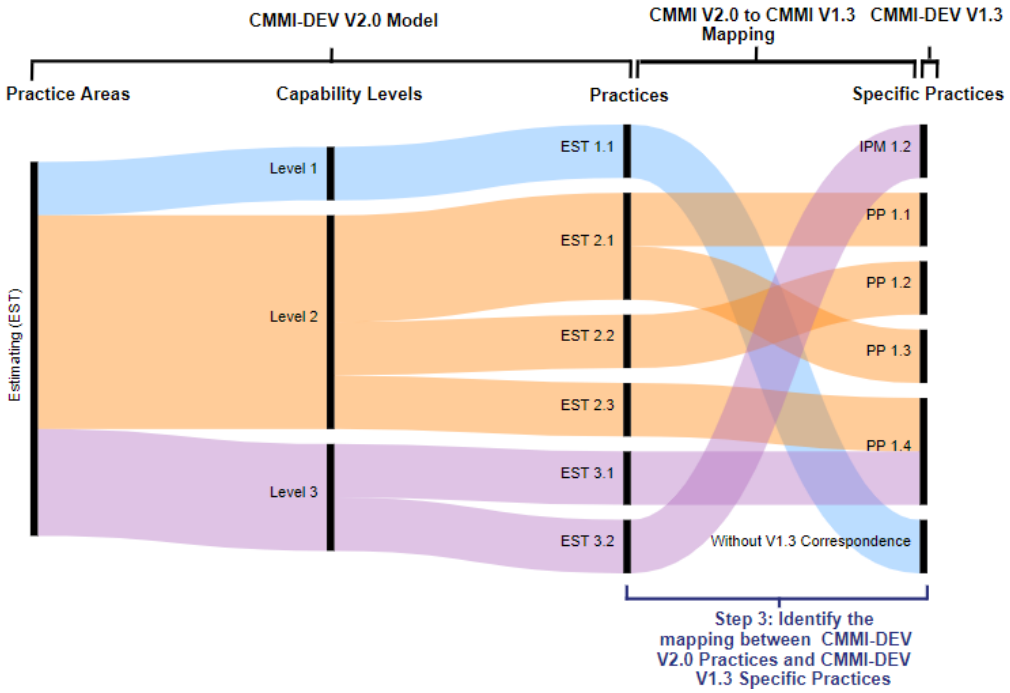

Figure 3. Summary of the mapping between CMMI-DEV V2.0 and CMMI-DEV V1.3 based on the example of the Estimating (EST) Practice Area (Figure 2, Step 3).

As Figure 3 shows, all the EST PA Practices, except EST 1.1 (Develop high-level estimates), map to CMMI-DEV V1.3 Specific Practices. For example, EST 2.2 (Manage estimates for the size of the solution), EST 2.3 (Based on size estimates, develop and record effort, duration, and cost estimates), EST 3.1 (Manage a recorded estimation method) and EST 3.2 (Use the organizational repository and process assets for estimating work) map, respectively, to the CMMI-DEV V1.3 PP 1.2 (Establish Estimates of Work Product and Task Attributes), PP 1.4 (Estimate Effort and Cost) and IPM 1.2 (Use Organizational Process Assets for Planning Project Activities) Specific Practices. Likewise, EST 2.1 (Manage the scope of what is being estimated) maps to two CMMI-DEV V1.3 PP 1.1 (Estimate the Scope of the Project) and PP 1.3 (Define Project Lifecycle Phases) Specific Practices.

We developed a visualization of the entire CMMI-DEV V2.0 and its mapping to CMMI-DEV V1.3, based on Figure 3, which is available at https://short.upm.es/6arew.

The mapping document mentions that the Governance (GOV) and Implementation Infrastructure (II) PAs are intentionally not mapped to CMMI-DEV V1.3 Generic Goals and Generic Practices because the Governance (GOV) and Implementation Infrastructure (II) PAs are not altogether compatible with previous CMMI versions [39].

The full analysis of the mapping document (following a similar process to the example in Figure 3) indicates that 147 out of the 186 CMMI-DEV V2.0 Practices can be mapped to CMMI-DEV V1.3 Specific Practices.

7 Consolidating the Agile Artifacts Associated with CMMI-DEV V2.0

This section describes Step 4 of Figure 2. To enrich CMMI-DEV V2.0 Practices with a set of Agile artifacts, we consolidate the results detailed in the previous sections, namely, the Agile artifacts associated with CMMI-DEV V2.0 PAs (Section 5); the Agile artifacts associated with CMMI-DEV V1.3 Specific Practices (Section 5), and the mapping between these two versions of CMMI-DEV (Section 6).

Figure 4 shows how Steps 1, 2, and 3 (bottom of Figure 4) bring to light the Agile artifacts (left- and right-hand sides of Figure 4) associated with the Estimating (EST) PA and its Practices.

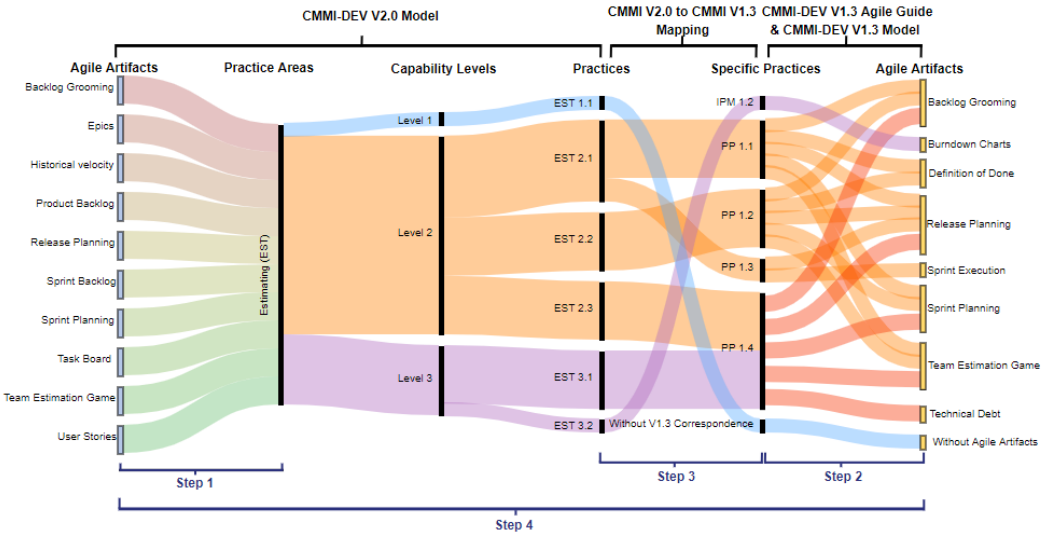

Figure 4. Relationship between Agile artifacts, CMMI-DEV V2.0, and CMMI-DEV V1.3 based on the example of the Estimating (EST) Practice Area (Figure 1, Steps 1, 2, and 3).

Figure 4 (left) shows the ten Agile artifacts associated with the Estimating (EST) PA. These associations are sourced from the content analysis of the CMMI V2.0 Model (see Table 2). Additionally, Figure 4 (right) shows the eight Agile artifacts associated with the EST 2.1, EST 2.2, EST 2.3, EST 3.1, and EST 3.2 Practices. These Practice-level associations are derived from Agile artifacts associated with their respective Specific Practices in CMMI-DEV V1.3 (see Figure 3 and Table 4).

Figure 5 provides a clear picture of the consolidation of the two sets of Agile artifacts shown in Figure 4 (left and right). Figure 5 (right) shows the consolidated list of the 14 Agile artifacts associated with Estimating (EST). We denote these Agile artifacts using three different colored boxes. The link to blue-colored boxes is derived from the agile guidance described in CMMI-DEV V2.0. Therefore, these six Agile artifacts are associated with Estimating (EST) at the *PA level only*. The source of the links to the yellow boxes is the agile guidance for CMMI-DEV V1.3 Specific Practices and their mapping to CMMI-DEV V2.0 Practices. In this case, there are four Agile artifacts associated at the *Practice level only*. Finally, the links to the green boxes are the product of both agile guidelines. In this case, there are four Agile artifacts associated with CMMI-DEV V2.0 at *both the PA level and the Practice level*.

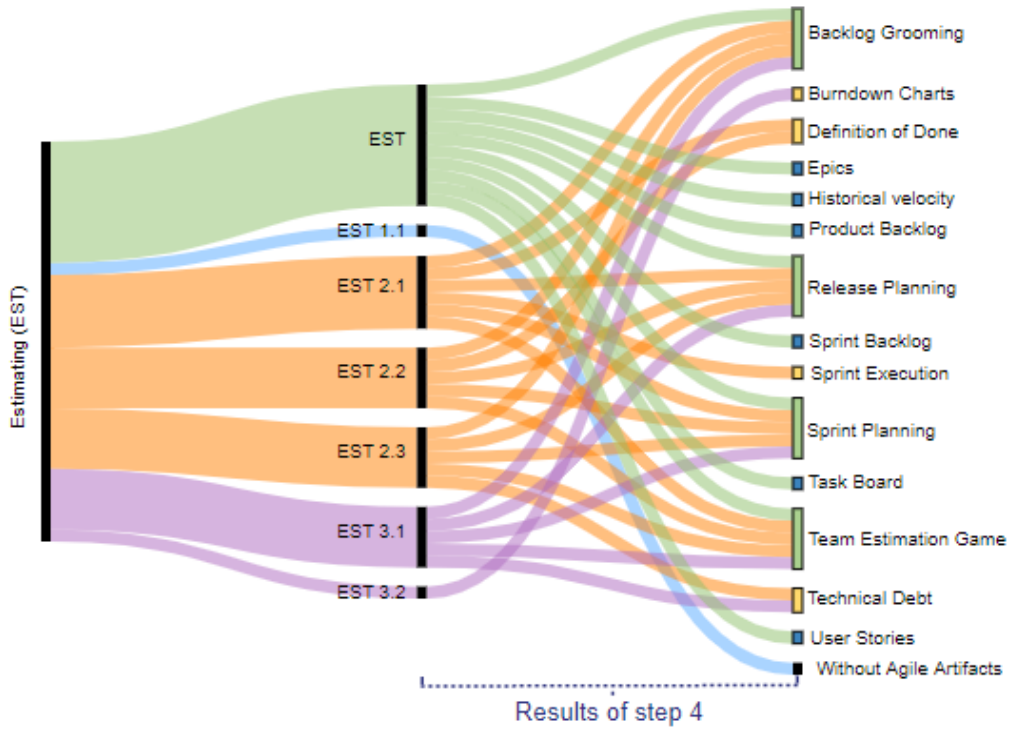

Figure 5. Consolidated list of Agile artifacts associated with the CMMI-DEV V2.0 Estimating (EST) Practice Area (Result of Figure 2, Step 4).

The colored boxes in Figure 5 provide traceability to the source of the association. Nevertheless, from a practical point of view, the important thing is the result of the association between Agile artifacts and CMMI-DEV V2.0 Practices.

In particular, the Release Planning, Sprint Planning, Backlog Grooming, and Team Estimation Game (the four green boxes) are identified by both guidelines [12] [37] for Estimating (EST). Therefore, these guidelines enrich the information provided by CMMI-DEV V2.0 at the PA level, adding details at the Practice level.

Additionally, the Sprint Execution, Definition of Done, Technical Debt, and Burndown Charts have been associated with the Estimating (EST) Practices by their mapping to CMMI-DEV V1.3 Specific Practices (the four yellow boxes). Therefore, they round out the guidance provided by CMMI-DEV V2.0, adding other Agile artifacts that contribute to the materialization of CMMI-DEV V2.0 Practices.

We generated a consolidated list of Agile artifacts associated with all CMMI-DEV V2.0 PAs and Practices following the process illustrated in Figures 4 and 5. Given the size of this list (31 Agile artifacts, 19 PAs, and 186 Practices), our overview has not been included here and is available at https://short.upm.es/uq5hj. This consolidated list is the answer to the question mark in Figure 1.

Table 6 shows an example of the consolidated list of Agile artifacts associated with the Monitor and Control (MC), Planning (PLAN), and Estimating (EST) PAs and their corresponding Practices, which are part of the "Planning & Managing Work" capability area of CMMI-DEV V2.0. The structure of this table is based on Table 2, where the associations between PAs and Agile artifacts are highlighted in blue. Unlike Table 1, the yellow cells represent associations between CMMI-DEV V2.0 Practices and the Agile artifacts that we have discovered in this step.

	Consolidated list of Agile artifacts pointed out by by the CMMI Institute																														
Practices Area and Practices	**Release Planning**	**Sprint Planning**	**Sprint Execution**	**Sprint Review/Demo**	**Retrospective**	**Daily Scrum**	**Backlog Grooming**	**Scrum of Scrums**	**DevOps**	**User Stories**	**Epics**	**Definition of Done**	**Code Review**	**Continuous Integration /Build**	**Team Estimation Game, planning**	**Pair Programming**	**Refactoring**	**Team Agreements**	**Technical Debt**	**Test Driven Dev.**	**Historical Velocity**	**Agile Process Definition**	**Product Backlog**	**Sprint Backlog**	**Release Plan**	**Task Board**	**Burndown Charts**	**Source Code & Executables**	**Acceptance Criteria/Test**	**Test plan, Cases & Result**	**Retrospective Data**
MC																															
MC 2.1																															
MC 2.2																															
MC 2.4																															
MC 3.1																															
MC 3.2																															
MC 3.4																															
PLAN																															
PLAN 2.1																															
PLAN 2.3																															
PLAN 2.4																															
PLAN 2.6																															
PLAN 2.7																															
PLAN 2.8																															
PLAN 3.1																															
PLAN 3.2																															
PLAN 3.3																															
EST																															
EST 2.1																															
EST 2.2																															
EST 2.3 EST 3.1																															
EST 3.2																															

Association by CMMI-DEV V2.0 Association by CMMI-DEV V1.3

Table 6. Association between Agile artifacts and CMMI-DEV V2.0 Planning & Managing Work Practice Areas and Practices.

8 Discussion

The results of this research reveal several outcomes to be considered by an Agile organization aiming to create a roadmap to align with CMMI-DEV V2.0. We set out and discuss the practical implications of these outcomes below.

- The Agile artifacts associated with most CMMI-DEV V2.0 PAs and Practices are: Sprint Execution, Sprint Planning, Release Planning, Sprint Review/Demo, Backlog Grooming, Retrospective, Product/Sprint Backlog, and Burndown Charts. These Agile artifacts can contribute to many CMMI-DEV V2.0 PAs and Practices. For example, Sprint Execution is associated with 15 CMMI-DEV V2.0 PAs. In practice, this result means that agile organizations should use a combination of their regular Agile artifacts to materialize CMMI-DEV V2.0 Practices. Therefore, the process improvement practices should be adopted as part of the current team activities.
- The Agile artifacts most consistently identified by CMMI V2.0 and V1.3 sources are related to the following PAs: Requirements Development and Management (RDM), Planning (PLAN), and Estimating (EST), and their Practices. Therefore, these PAs appear to be more consolidated insofar as their related Agile artifacts are supported by both sources. In the same vein, most Practices in the Verification and Validation (VV), Requirements Development and Management (RDM), Estimating (EST), Risk and Opportunity Management (RSK), Product Integration (PI), and Peer Reviews (PR) PAs have been associated with Agile artifacts. This is positive for agile organizations that intend to deliver the business value of such PAs (and their respective Practices). This result is not surprising, as most of the guidelines are based on Scrum and XP approaches, which are by nature primarily focused on project/product software development management. This result has two practical implications. On the one hand, these CMMI V2.0 Practices (from EST, PLAN, PI, PR, RSK, and VV) can be straightforwardly materialized (since the available guidelines are more consolidated). For instance, EST Practices can be performed during sprint planning and grooming, and the generated records are then posted on the sprint boards. On the other hand, CMMI embodies practices that most adopted Agile approaches were not intended to address. Therefore, organizations will need to explore broader Agile approaches to address the entire CMMI V2.0.
- Our content analyses have not provided Agile artifacts to enrich the Process Quality Assurance (PQA), Process Management (PCM), Decision Analysis and Resolution (DAR), and Causal Analysis and Resolution (CAR) Practices. Accordingly, these PAs only have Agile artifacts associated at PA level. Therefore, the implementation effort will probably be bigger than for other PAs that have been enriched. In practice, this result could mean that Agile organizations should opt to track the results generated from internal audits performed, for instance, in the sprint review, peer review activities, and retrospectives in order to materialize PQA practices. They should also systematically collect and analyze information about improvements needed across the organization, for instance, by sharing the data collected in retrospectives and creating communities of practice to address problems shared by more than one team. This can help to materialize PCM Practices. In the same vein, the DAR and CAR Practices should be adopted during sprint planning, review, and retrospectives.
- The CMMI Institute has not provided agile guidelines on Governance (GOV), Implementation Infrastructure (II), and Organizational Training (OT). Although implementation effort does not depend on the extent of guidelines alone, the implementation of such PAs in Agile settings is likely to consume more resources than are required for other CMMI-DEV V2.0 PAs, where guidelines are provided at either the Practice Area or Practice levels. In practice, Agile organizational frameworks are an alternative for adopting an Agile-CMMI approach at the organizational level. For instance, PMI [40] details the difference between Lean and traditional governance, which can help to materialize GOV Practices. In the same vein, SAFe [41] provides guidelines to achieve organizational agility and create a continuous learning culture that can help to materialize CMMI V2.0 GOV, II and OT Practices.

Based on the evidence existing in the literature, we are unable to estimate what coverage Agile artifacts provide for CMMI practices. We can infer that a combination of Agile artifacts may suffice to achieve CMMI practices, since most context-specific sections start by *[PA] is typically performed in an agile project using Scrum* [37]. One exception to this is RDM, which states that *RDM practices add early*

consideration of other requirement types beyond the user stories typically used by an Agile project using Scrum [37]. One of the analyzed documents claims that CMMI adoption in Agile settings should not mean *do CMMI*; rather it should mean *do Agile better* [12]. Again, this suggests that some adjustments are needed to align both approaches, but the documents do not provide enough information to quantify the extent of such adjustments.

From a broader perspective, we should note that Agile approaches usually leave fewer traces of the work done than traditional approaches. As a result, agile organizations seeking to align with CMMI often struggle to provide sufficient evidence during the appraisal. However, this evidence does exist and can be managed more systematically by agile organizations that adopt CMMI V2.0 MPM, PAD, and CM Practices, among others, to promote the definition of metrics, process assets, and configuration items, respectively. For instance, the collection and analysis of sprint burndown charts is a valuable resource for making measurements and recording performance, whereas other metrics automatically collected from continuous integration servers can help to materialize MPM Practices. Additionally, a regular agile team performs tasks that can evidence CMMI Practices during each iteration. With regard to OT and PAD, for instance, agile teams share knowledge using wikis or by creating communities of practice. Also, cross-cutting CM, PCM, MPM, PQA PAs agile teams commonly use tools like Git, or Bitbucket to control versioning documents and code, and other tools like Redmine, Confluence or VersionOne to manage a repository of measurements, guidelines, and work environment standards, which can be particularly helpful for recording information for evidencing Maturity Levels 4 and 5.

In the agile community, there is a common belief that the alignment with CMMI reduces agility [42]. Nevertheless, this may be a mere perception. For instance, it was reported that agile processes worked more effectively after adopting CMMI-DEV V1.3 Maturity Level 5 [28]. Therefore, it is essential to collect and analyze specific metrics to monitor agility during alignment with CMMI V2.0. Such metrics should go beyond velocity or time to release and account for variables like software type, quality criteria, and team particularities. If an organization continuously controls agility indicators at organizational level, or at least at team level, the risk of losing agility during the alignment with CMMI can be mitigated.

Figure 6 shows an overview of CMMI-DEV V2.0 PAs according to the extent of the Agile artifacts brought to light by this research. All groups, except the last, represent PAs with Agile artifacts. The PAs with Agile artifacts are grouped according to the percentage of Practices with identified Agile artifacts. The first group shows PAs with Agile artifacts, where 66% or more of their Practices have related Agile artifacts. The second group shows the PAs with agile practices where from 34% to 65% of their Practices have related Agile artifacts. The third group shows the PAs with Agile artifacts where less than 34% of their Practices have associated Agile artifacts. The fourth group shows the PAs with Agile artifacts where none of their Practices have related Agile artifacts. This information can be used by organizations aligning Agile with CMMI V2.0 to create a roadmap suited to their pursued business value.

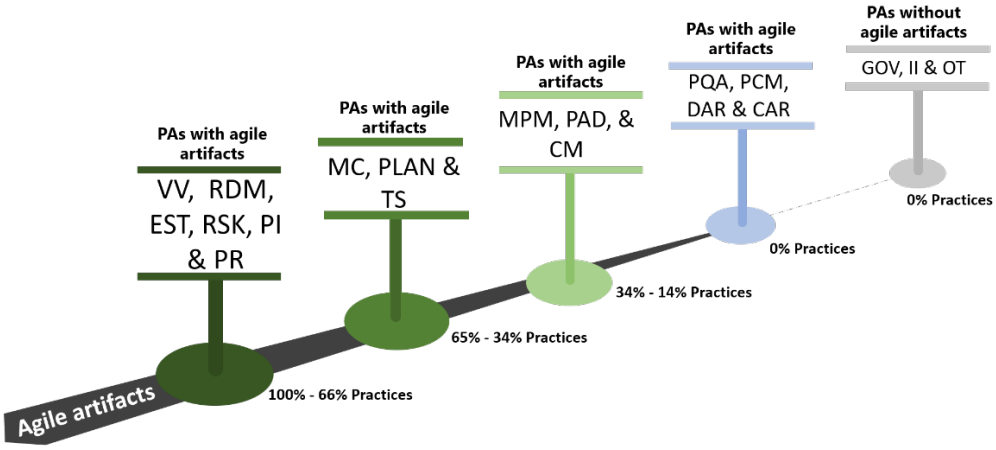

Figure 6. CMMI-DEV V2.0 PAs according to the extent of the Agile artifacts pointed out by the CMMI Institute.

Additionally, agile organizations designing an alignment roadmap can use the results detailed throughout this article to adopt CMMI-DEV V2.0 from scratch or as a transition from the previous version. On the one hand, agile organizations can use the consolidated list of Agile artifacts discussed in Section 6 and available at https://short.upm.es/uq5hjj to quickly check the Agile artifacts now in place against the agile artifact identified for their pursued CMMI-DEV V2.0 Practices. On the other hand, agile organizations transitioning from CMMI-DEV V1.3 to CMMI-DEV V2.0 can use the mapping between both models discussed in Section 5 and available at https://short.upm.es/6arew to guide the transition.

Note that, as mentioned in the introduction, the results of our research do not constitute a two-way mapping between Agile artifacts and CMMI.DEV V2.0 Practices. Far from it, the two approaches are orthogonal and complementary, as they both provide means to help organizations deliver business value [27]. Neither are our results a prescription of Agile artifacts for implementing CMMI-DEV V2.0. Given the wide range of projects and technologies, as well as the idiosyncrasy of each organization, it is impossible to generate a one-size-fits-all solution. Therefore, each organization should determine, driven by its business needs, how to materialize CMMI-DEV V2.0 Practices in its agile context. The results generated in this research are a starting point to help each organization design a specific and ad hoc roadmap to rise to this challenge.

9 Conclusion and Future Work

Unlike the literature that provides an ad hoc mapping of Agile artifacts with CMMI-DEV V1.3 Specific Practices, this research identifies the associations provided by CMMI Institute sources and focuses on CMMI-DEV V2.0. Our study reveals that the adoption of an agile artifact can help to materialize many CMMI-DEV V2.0 Practices in a hybrid (CMMI and Agile) context. Likewise, the adoption of a CMMI-DEV V2.0 Practice entails the use of many Agile artifacts.

The consolidated list of Agile artifacts associated with CMMI-DEV V2.0 generated in this research specify the agile practices pointed out by the CMMI Institute. However, this does not mean that other Agile artifacts are not applicable. Note that the results of our research do not constitute a two-way mapping between Agile artifacts and CMMI.DEV V2.0 Practices. Far from it, the two approaches are orthogonal and complementary, as they both provide best means to help organizations deliver business value [27].

The key findings of our research can be summarized as follows. We have rounded out the Agile guidance for 12 out of 19 CMMI-DEV V2.0 PAs. In particular, we have linked Agile artifacts to 72 out of the 186 CMMI-DEV V2.0 Practices, which represent 41% of the Practices at CMMI-DEV V2.0 Capability Levels 2 and 3, which are, in fact, the maturity levels at which most organizations are certified [5]. The detailed information provided in this paper could help to draft an action plan to undertake a CMMI-DEV V2.0-based process improvement initiative in the two scenarios mentioned in Section 1:

- CMMI-DEV V1.3-certified agile organizations can map their current agile and CMMI-DEV V1.3 practices to CMMI-DEV V2.0 more easily. In fact, such organizations can use the results discussed in Section 5 and available at https://short.upm.es/6arew to exploit their current practices for the transition to CMMI-DEV V2.0
- Agile organizations intending to adopt CMMI-DEV V2.0 can be quicker at checking the Agile artifacts now in place at their organization against the consolidated list of Agile artifacts derived from literature (see https://cutt.ly/vtJzHij discussed in Section 5).

Note that, because of the wide range of projects, products and technologies and the idiosyncrasy of each organization, it is impossible to generate a one-size-fits-all solution. Therefore, each organization should determine, driven by its business needs, how to materialize CMMI V2.0 Practices in its agile context.

On the other hand, these results are based on document content analysis, and no industrial validation has been carried out so far. We are aware that practical implementations would provide very interesting insight in order to strengthen our findings. Nevertheless, this research is a preliminary starting point based on the agile practices derived from the analyzed documents, which compile the experience of different CMMI consultants working on a variety CMMI appraisals in Agile settings [12].

Future work should address different lines of action. First, further validation is needed to reinforce our results; expert panels and case studies are an immediate option. Researchers might also want to focus on addressing the gap highlighted in this paper: 61% of CMMI-DEV V2.0 Practices are not explicitly associated with Agile artifacts. This would require a) identifying Agile artifacts that mirror the GOV, II and OT PAs, the first two of which are mandatory for achieving any CMMI-DEV V2.0 maturity or capability level; and b) determining which Agile artifacts help to materialize CMMI-DEV V2.0 Capability Levels 4 and 5.

It would be particularly interesting to explore how Agile artifacts could address the CMMI-DEV V2.0 GOV PA and its Practices. Traditional (command-and-control) governance can generate organizational conflicts or "organizational type mismatch" in Agile contexts [27], which undermine the productivity, quality and morale of agile software development teams [43]. Therefore, this is a very important issue for Agile-CMMI V2.0 alignment.

Additionally, future work could focus on evaluating the quality of the Agile guidance that is part of the CMMI V2.0 model. Thus, it would be possible to determine to what extent the available PA-level guidance can reduce the effort required to arrive at the effective alignment of both approaches.

Another possible line of research is to generate guidance for other Agile contexts, as, although Scrum is widely used, there is a steady growth in the application of other frameworks related to the scaled adoption of Agile, like SAFe or PMI. These frameworks could help to materialize the CMMI-DEV V2.0 GOV, II, and OT PAs, as they detail values, principles, roles and responsibilities in order to assure transparent, high-performance relationships among agile team members and encourage a continuous learning culture [40][41].

10 Declarations

Funding: Not applicable
Competing interests: The authors declare that they have no competing interest
Availability of Data and Material: All the data supporting the conclusions of this article are available at the external link https://short.upm.es/v32on
Code availability: Not applicable

11 Acknowledgements:

This work was supported in part by the ANID PFCHA/DOCTORADO BECAS CHILE/2018 - 72190378.

References

[1] E.M. Schön, J. Thomaschewski, M.J. Escalona, Agile Requirements Engineering: A systematic literature review, Comput. Stand. Interfaces. 49 (2017) 79–91. https://doi.org/10.1016/j.csi.2016.08.011.

[2] R. Hoda, N. Salleh, J. Grundy, The Rise and Evolution of Agile Software Development, IEEE Softw. 35 (2018) 58–63. https://doi.org/10.1109/MS.2018.290111318.

[3] Digital.ai, 15th State of Agile Report, Digital.Ai. (2021) 1–23. https://stateofagile.com/#ufh-i-661275008-15th-state-of-agile-report/7027494 (accessed November 15, 2021).

[4] ISACA, CMMI V2.0 Performance Report Summary, 2020. https://cmmiinstitute.com/getattachment/738104c0-a6f0-4e1c-8bbe-35076b75f36e/attachment.aspx.

[5] C. Institute, CMMI Adoption Trends, 2019. https://cmmiinstitute.com/resource-files/public/cmmi-adoption-trends-2018-year-end-update (accessed November 15, 2021).

[6] P. Sharma, A.L. Sangal, Building a hierarchical structure model of enablers that affect the software process improvement in software SMEs—A mixed method approach, Comput. Stand. Interfaces. 66 (2019) 103350. https://doi.org/10.1016/j.csi.2019.04.009.

[7] J. Masso, F.J. Pino, C. Pardo, F. García, M. Piattini, Risk management in the software life cycle: A systematic literature review, Comput. Stand. Interfaces. 71 (2020).

https://doi.org/10.1016/j.csi.2020.103431.

[8] D. Proença, J. Borbinha, Formalizing ISO/IEC 15504-5 and SEI CMMI v1.3 – Enabling automatic inference of maturity and capability levels, Comput. Stand. Interfaces. 60 (2018) 13–25. https://doi.org/10.1016/j.csi.2018.04.007.

[9] ISACA, CMMI Institute, Take Organizational Performance to the Next Level with CMMI V2.0, 2020. https://cmmiinstitute.com/getattachment/f182a1a8-1ebf-4737-9d40-4c2e4b9456a1/attachment.aspx.

[10] J.J. Esther, Agile methodology tailoring Under CMMI, Wilmington University (Delaware), 2019. https://search.proquest.com/openview/fba1ab0e990c6def431748159b58bd44/1.pdf?pq-origsite=gscholar&cbl=18750&diss=y.

[11] D. Carlson, E. Soukup, Combining Agile Practices with CMMI Process Areas A Case Study, CrossTalk. (2017) 17–21. https://apps.dtic.mil/sti/pdfs/AD1057246.pdf.

[12] J. Dalton, R. Timmerman, L. Adkins, K. Botula, N. Potter, D. Torrens, M. Glover, A Guide to Scrum and CMMI: Improving Agile Performance with CMMI, 2016. https://cmmiinstitute.com/resource-files/public/marketing/document/a-guide-to-scrum-and-cmmi®-improving-agile-perfor.

[13] IASACA, When was CMMI V2.0 released?, (2020). https://isaca.force.com/support/s/article/When-was-CMMI-V2-0-released-1598331746732 (accessed October 15, 2021).

[14] ISACA, How is CMMI V2.0 different from V1.3?, (2020). https://isaca.force.com/support/s/article/How-is-CMMI-V2-0-different-from-V1-3-1598331744955 (accessed October 15, 2021).

[15] V. Henriquez, A.M. Moreno, J.A. Calvo-Manzano, T. San Feliu, Agile-CMMI Alignment : CMMI V2.0 Contributions and To-dos for Organizations, Computer. https://doi.org/Accepted for publication on November 2020.

[16] C.J. Torrecilla-Salinas, J. Sedeño, M.J. Escalona, M. Mejías, Agile, Web Engineering and Capability Maturity Model Integration : A systematic literature review . R, Inf. Softw. Technol. 71 (2016) 92–107. https://doi.org/10.1016/j.infsof.2015.11.002.

[17] L.F. Chagas, D.D. De Carvalho, A.M. Lima, C. Alessandra, L. Reis, Systematic Literature Review on the Characteristics of Agile Project Management in the Context of Maturity Models, in: Softw. Process Improv. Capab. Determ. SPICE 2014. Commun. Comput. Inf. Sci., 2014: pp. 177–189. https://doi.org/https://doi.org/10.1007/978-3-319-13036-1_16.

[18] F.S. Silva, F.S.F. Soares, A.L. Peres, I.M. De Azevedo, A.P.L.F. Vasconcelos, F.K. Kamei, S.R. De Lemos Meira, Using CMMI together with agile software development : A systematic review, Inf. Softw. Technol. 58 (2015) 20–43. https://doi.org/10.1016/j.infsof.2014.09.012.

[19] M. Palomino, A. Dávila, K. Melendez, M. Pessoa, E. De Posgrado, P. Universidad, Agile Practices Adoption in CMMI Organizations : A Systematic Literature Review, in: Int. Conf. Softw. Process Improv., 2017: pp. 57–67. https://doi.org/10.1007/978-3-319-48523-2_6.

[20] V. Henriques, M. Tanner, A Systematic Literature Review of Agile and Maturity Model Research, Interdiscip. J. Information, Knowledge, Manag. 12 (2017) 53–73. http://www.ijikm.org/Volume12/IJIKMv12p053-073Henriques3025.pdf.

[21] CMMI Institute, CMMI V2.0 Practice Mapping, 2018. https://cmmiinstitute.com/resource-files/public/v2-0-materials/cmmi-v2-0-to-v1-3-practice-mapping.

[22] S. Sreenivasan, K. Kothandaraman, Improving processes by aligning Capability Maturity Model Integration and the Scaled Agile Framework ®, WILEY. 38 (2019) 42–51. https://doi.org/10.1002/joe.21966.

[23] M. Denscombe, The Good Research Guide: For small-scale social research projects, McGraw-Hill Education (UK), 2014. https://doi.org/9780335264704.

[24] S. Pulparambil, Y. Baghdadi, Service oriented architecture maturity models: A systematic literature review, Comput. Stand. Interfaces. 61 (2019) 65–76. https://doi.org/10.1016/j.csi.2018.05.001.

[25] M.L. Sanchez-Gordon, A. de Amescua, R. V. O'Connor, X. Larrucea, A standard-based framework to integrate software work in small settings, Comput. Stand. Interfaces. 54 (2017) 162–175. https://doi.org/10.1016/j.csi.2016.11.009.

[26] I. Monarch, D. Goldenson, P. Capell, J. Wessel, Tailoring and Combining the CMMI-ACQ and Quality

Models to Improve the Military's Requirements Process, 2008. https://apps.dtic.mil/sti/pdfs/ADA632545.pdf.

[27] J. Dalton, Great Big Agile: An OS for Agile Leaders, Apress, 2018. https://www.apress.com/gp/book/9781484242056.

[28] D. Sharma, N. Nishi, L. Djindo, L. Theron R., Agile 5 using high maturity CMMI practices to improve agile processes and achieve predictable results, CrossTalk. 29 (2016) 32–35. Agile 5 using high maturity CMMI practices to improve agile processes and achieve predictable results.

[29] CMMI Institute, CMMI® and Agile, (n.d.). https://cmmiinstitute.com/agile (accessed February 11, 2020).

[30] A. Ferdinansyah, B. Purwandari, Challenges in Combining Agile Development and CMMI : A Systematic Literature Review, in: 10th Int. Conf. Softw. Comput. Appl., 2021: pp. 63–69. https://doi.org/https://doi.org/10.1145/3457784.3457803.

[31] A. Bahaa, A.S. Abd, Y. Mostafa, Implementing Project Management Category Process Areas of CMMI Version 1.3 Using Scrum Practices, and Assets, Int. J. Adv. Comput. Sci. Appl. 7 (2016) 243–252. https://doi.org/10.14569/ijacsa.2016.070234.

[32] C.J. Torrecilla-Salinas, J. Sedeño, M.J. Escalona, M. Mejías, An Agile approach to CMMI-DEV levels 4 and 5 in Web development projects, in: 25th Int. Conf. Inf. Syst. Dev. ISD 2016, Katowice, Poland, 2016: pp. 137–149. http://aisel.aisnet.org/isd2014/proceedings2016/ISDevelopment/2.

[33] C.J. Torrecilla-Salinas, T. Guardia, O. De Troyer, M. Mej, J. Sede, NDT-Agile : An Agile, CMMI-Compatible Framework for Web Engineering, in: Int. Conf. Softw. Process Improv. Capab. Determ., 2017: pp. 3–16. https://doi.org/10.1007/978-3-319-67383-7.

[34] S. Kawamoto, J.R. De Almeida, Scrum-DR: An extension of the scrum framework adherent to the capability maturity model using design rationale techniques, 2017 Chil. Conf. Electr. Electron. Eng. Inf. Commun. Technol. CHILECON 2017 - Proc. 2017-Janua (2017) 1–7. https://doi.org/10.1109/CHILECON.2017.8229530.

[35] R. Albuquerque, R. Fontana, A. Malucelli, S. Reinehr, Agile Methods and Maturity Models Assessments: What's Next?, in: Eur. Conf. Softw. Process Improv., 2019: pp. 619–630. https://doi.org/10.1016/j.csi.2013.08.001.

[36] C. Institute, GlobalLogic Brings Predictability to Agile with CMMI, 2019. https://cmmiinstitute.com/resource-files/public/globallogic-case-stuy.

[37] CMMI Institute, CMMI Model V2.0, 2018. https://cmmiinstitute.com/resource-files/public/cmmi-v2-0-development-model.

[38] SEI, CMMI for Development, Version 1.3, 2010. https://resources.sei.cmu.edu/asset_files/TechnicalReport/2010_005_001_15287.pdf.

[39] CMMI Institute, CMMI V2.0 to V1.3 Detailed Practice Mapping (Excel), (2018). https://cmmiinstitute.com/resource-files/public/v2-0-materials/cmmi-v2-1-to-v1-3-detailed-practice-mapping-(excel (accessed November 5, 2019).

[40] S. Ambler, G. Little, M. Lines, V. Mocanu, D. Gagnon, M. Richardson, J. Barnes, S. Ambler, Lean vs . Traditional IT Governance From the Disciplined Agile Blog by Scott Ambler, Glen Little, Mark Lines, Valentin Mocanu, Daniel Gagnon, Michael Richardson, Joshua Barnes, (2020) 1–4. https://www.projectmanagement.com/blog/blogPostingView.cfm?blogPostingID=61873&thisPageURL=/blog-post/61873/Lean-vs--Traditional-IT-Governance#_=_ (accessed November 15, 2021).

[41] Scaled Agile Framework, SAFe 5.0 Framework - SAFe Big Picture, (n.d.). https://www.scaledagileframework.com/# (accessed July 1, 2020).

[42] F. Selleri, F. Santana, F. Soares, A. Lima, I. Monteiro, D. Azevedo, A. Paula, L.F. Vasconcelos, F. Kenji, S. Romero, D.L. Meira, Using CMMI together with agile software development : A systematic review, Inf. Softw. Technol. 58 (2015) 20–43. https://doi.org/10.1016/j.infsof.2014.09.012.

[43] S. Ambler, G. Little, M. Lines, V. Mocanu, D. Gagnon, M. Richardson, J. Barnes, K. Birk, K. Boedker, M. Griffiths, Should You Govern Agile Teams Via a Traditional Strategy? – Disciplined Agile (DA), (2018). https://www.pmi.org/disciplined-agile/people/governing-agile-teams (accessed September 20, 2020).