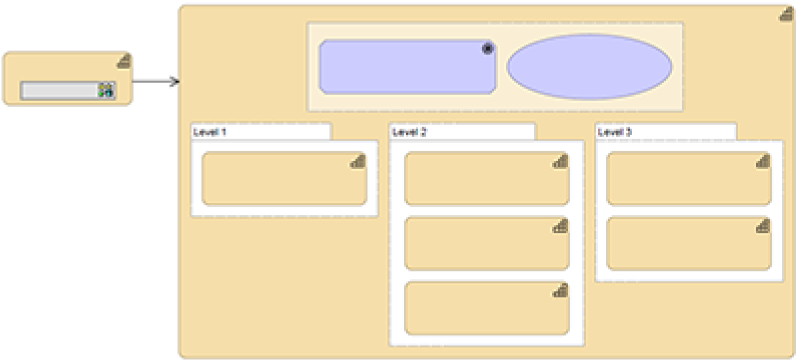


A Reference Model for Migrating from CMMI-DEV v1.3 to CMMI v2.0

Mariana Cardoso Lourenço dos Santos Silva

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Miguel Leitão Bignolas Mira da Silva
Engª Ana Margarida Gomes Rolão Gonçalves

Examination Committee

Chairperson: Prof. Miguel Nuno Dias Alves Pupo Correia
Supervisor: Prof. Miguel Leitão Bignolas Mira da Silva
Member of the Committee: Prof. Alberto Manuel Rodrigues da Silva

October 2019

Acknowledgments

First, I would like to express my gratitude to my dissertation supervisors, Prof. Miguel Mira da Silva and Margarida Gonçalves, for all the guidance, insight, and sharing of knowledge that made this thesis possible.

Secondly, I would also like to express my gratitude to the people who collaborated in the demonstration, for their availability and for providing me valuable information and feedback that enriched the demonstration of our proposal. To Maarten Steen from BiZZdesign, for providing me an academic license for BiZZdesign Enterprise Studio, this tool was very helpful to develop our models. And to Nuno Seixas, for the interest in my thesis, the CMMI knowledge shared with me, and the valuable feedback provided.

I would like to thank my family for their unconditional support and understanding throughout all these years. Specially, to my godmother and uncle for their availability, for giving me advice, and providing me so many experiences that truly enriched my life.

To my boyfriend, that always supported me and was there for me whenever I was stressing out, for being caring and providing me emotional support.

Last but not least, to my closest friends, that accompanied me through this journey in Técnico, that were always available to help me solve any problem and share their advice. They were a crucial part to survive these stressful times, giving me encouragement and always being there whenever I needed a break.

To each and every one of you – Thank you.

Abstract

In today's high-technology business environment, the success of an organization is highly influenced by the functionality and quality of the software they use and develop. The challenge is to deliver reliable software on time and on budget. CMMI helps companies improve their software development processes and, although the benefits are clear, the CMMI textual reference models are complicated. With the new version of CMMI, new concepts and relationships were introduced, thus we propose a CMMI v2.0 Reference Model in ArchiMate to facilitate the migration for companies that are already CMMI-DEV v1.3 accredited. To guide our work we used the Design Science Research Methodology and the utility of the model is demonstrated in a real-world organization that is CMMI-DEV v1.3 accredited and needs to migrate to CMMI v2.0. The demonstration is based on CMMI models in ArchiMate, together with models of the AS-IS and TO-BE enterprise architecture of the organization. To validate the proposed reference model and the demonstration, we used questionnaires and interviews to CMMI experts and practitioners, well as well-known techniques to evaluate Design Science artifacts.

Keywords

Reference Model; Capability Maturity Model Integration; Enterprise Architecture; ArchiMate.

Resumo

Atualmente, neste ambiente de negócios tão tecnológico, o sucesso de uma organização é bastante influenciado pela funcionalidade e qualidade do software que utilizam e desenvolvem. O desafio é fornecer software confiável dentro do prazo e do orçamento. O CMMI ajuda as empresas a melhorar os seus processos de desenvolvimento de software e, embora os benefícios sejam claros, os modelos de referência textual do CMMI são complicados. Com a nova versão do CMMI, novos conceitos e novos relacionamentos foram introduzidos. Deste modo, propomos um modelo de referência do CMMI v2.0 em ArchiMate, para facilitar a migração para empresas que já têm uma acreditação em CMMI-DEV v1.3. Para orientar o nosso trabalho, usámos a metodologia de pesquisa em Design Science. A utilidade do modelo é demonstrada numa organização real que é acreditada em CMMI-DEV v1.3 e precisa de migrar para o CMMI v2.0. A demonstração é baseada nos modelos de CMMI em ArchiMate, juntamente com modelos da arquitetura atual e futura da organização. Para validar o modelo de referência proposto e a demonstração, para além de técnicas conhecidas para avaliar artefatos de Design Science, utilizámos questionários e entrevistas com especialistas e profissionais de CMMI.

Palavras Chave

Modelos de Referência; Capability Maturity Model Integration; Arquitetura Empresarial; ArchiMate.

Contents

List of Figures

List of Tables

Acronyms

ADM Architecture Development Method

CMMI Capability Maturity Model Integration

CMMI v2.0 CMMI version 2.0

CMMI-DEV v1.3 CMMI for Development version 1.3

COBIT Control Objectives for Information and Related Technologies

DS Design Science

DSRM Design Science Research Methodology

EA Enterprise Architecture

IS Information Systems

IT Information Technology

ITIL Information Technology Infrastructure Library

PR Peer Reviews

ROI Return on Investment

SPI Software Process Improvement

TIPA Tudor's ITSM Process Assessment

TOGAF The Open Group Architecture Framework

VAL Validation

VER Verification

VV Verification and Validation

1

Introduction

Contents

Nowadays, in the current business practice, an integrated approach to business and Information Technology (IT) is indispensable. The propagation of IT is an enterprise reality and the success of the business is vastly influenced by the functionality and quality of the software companies use and develop.

To follow the best practices known in the industry, organizations increasingly focus on redesigning their software processes aiming at a more reliable software that fits its purpose and is delivered to customers on time and on budget [6].

These Software Process Improvement (SPI) initiatives help organizations achieve their strategic goals while also being aligned with their business goals. They make processes the focal point, which is important since many of the problems in software development companies are caused by faulty processes rather than by people [6].

SPI has become an indispensable tool for software engineers and managers to accomplish their goals since it provides a Return on Investment (ROI) to the organization. It helps software companies deliver the agreed software on time and on budget and improves the quality of the delivered software, while reducing the cost of development and improving customer satisfaction [6].

To support SPI, there are several standards and frameworks available. In this thesis, we are using Capability Maturity Model Integration (CMMI).

CMMI is an internationally recognized model, used worldwide by thousands of organizations. According to CMMI Institute, this framework consists of "a set of best practices that enable businesses to improve performance of their key business processes" [7]. At its core, CMMI provides organizations a roadmap for process improvement through the development and maintenance of capabilities.

Benefits of this framework include improvements in several categories such as cost, schedule, productivity, quality, customer satisfaction and ROI [7, 8].

Although CMMI has clear benefits, studies show that the program is expensive and takes a lot of time and resources to implement [9–11]. One reason for it is that CMMI models are complicated. The existing textual reference models contain very extensive text, various technical concepts, and numerous relationships between different CMMI practices. Additionally, there are many different concepts in the two most recent versions of CMMI.

With the new version, CMMI version 2.0 (CMMI v2.0), if a company that is CMMI for Development version 1.3 (CMMI-DEV v1.3) accredited wants to maintain the accreditation, they need to migrate to CMMI v2.0. Hence, the main objective is to facilitate the migration from CMMI-DEV v1.3 to CMMI v2.0.

This thesis is a continuation of Valverde et al. [12] work. The author proposed a graphical reference model for CMMI-DEV v1.3 using ArchiMate as the chosen Enterprise Architecture (EA) modelling language, to reduce the high perceived complexity of CMMI by its users.

To address CMMI textual reference models being complicated and provide organizations a better understanding of CMMI v2.0, we propose a reference model of CMMI v2.0 using the EA modelling language, ArchiMate. This language allows us to describe and visualize our structure in a clear and simple way, thus we represented both concepts and relationships of CMMI v2.0 in ArchiMate's graphical elements.

1.1 Research Methodology

The approach chosen to guide this work was the Design Science Research Methodology (DSRM). The DSRM is an iterative methodology which combines principles, practices, and procedures required to carry a Design Science (DS) research. It provides guidance for research in Information Systems (IS) and other disciplines, as well as a mental model to present and evaluate DS research in IS [1, 13].

The main goal of DS in IS research is to create and evaluate IT artifacts intended to support the solution for the identified problems [1]. These artifacts can be constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices) and instantiations (implemented and prototype systems) [13].

In this thesis, the artifacts that we are going to create and evaluate are models and constructs, by models we are referring to the metamodel of CMMI and by constructs, the mapping of CMMI v2.0 in ArchiMate [1].

The DSRM process (Fig. 1.1) includes the following phases [1]:

1. **Problem identification and Motivation:** Define the specific research problem, justify the value of a solution and motivate the researcher to investigate the answer;
2. **Define the objectives of a solution:** Infer the objectives of a solution for the defined problem and the knowledge of what is achievable;
3. **Design and development:** Create the artifactual solution. Such artifacts can be constructs, models, methods or instantiations created to address the designated problem;
4. **Demonstration:** Demonstrate the efficacy of the artifacts to solve the problem;
5. **Evaluation:** Examine and measure how well the artifacts support the problem's solution, comparing the objectives to the results collected from use of the artifacts in the demonstration;
6. **Communication:** Communicate the problem, the artifacts, and the design, considering its relevance, utility, novelty, and effectiveness to researchers and other relevant audiences.

Being an iterative process, the DSRM allows to iterate many times through various phases and, in each iteration, obtain frequent and valuable feedback for the design process and its incremental improvement. In this thesis, two iterations of the design and development phase were done. We developed a first version of our reference model, then, after obtaining feedback from CMMI experts and practitioners, we improved it.

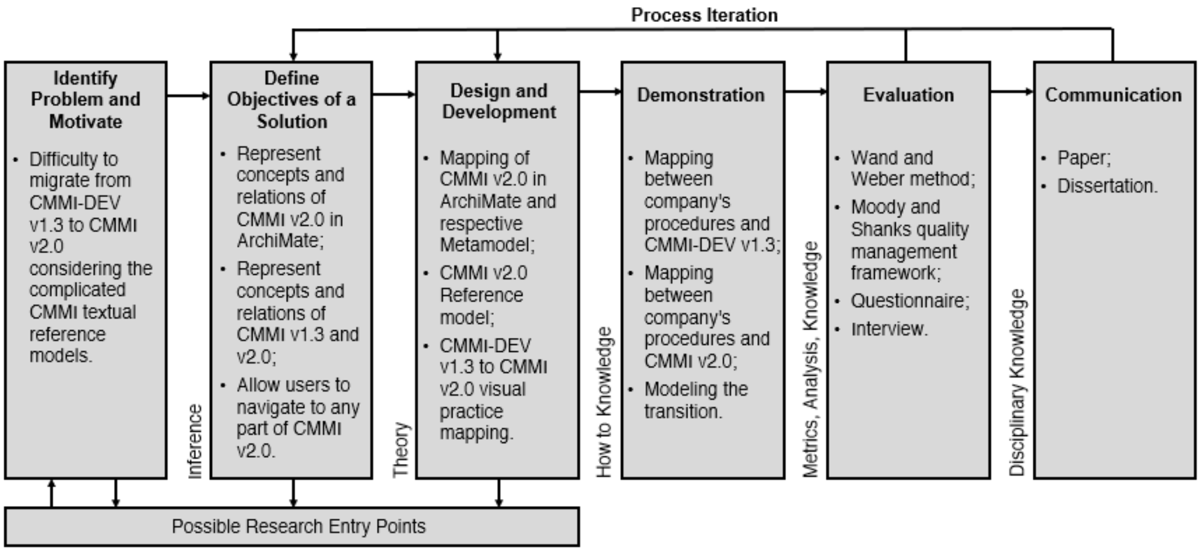


Figure 1.1: DSRM mapped to this thesis (Adapted from [1])

1.2 Research Outline

As described in section 1.1, we chose the DSRM to conduct this work. Therefore, the structure of this thesis is highly influenced by this methodology.

Chapter 1 begins to explain the thesis theme, followed by the chosen research methodology and structure of the research. Chapter 2 contains the motivation for this work and the statement of our research problem. Chapter 3 is the theoretical background, where we describe the main concepts necessary to understand this thesis. Chapter 4 contains the related work, which consists of an analysis of already existing solutions related to this thesis' context. Chapter 5 is the design and development phase, where we present our research proposal, well as the main objectives we want to achieve with its use. Chapter 6 is the demonstration of the proposed solution in a real-world organization. Chapter 7 is the evaluation of the proposed solution. Lastly, Chapter 8 concludes with an overview of the work that was done, its communication, contributions, limitations, and future work.

2

Research Problem

This chapter corresponds to the first phase of the DSRM, introduced in section 1.1. Here, we define the problem statement, justify the value of our proposal, and present a motivation for our thesis work.

In today's high-technology business environment, an integrated approach to business and IT is indispensable to face the challenges of the changing global business scene [6].

SPI programs have attracted much attention in research and practice due to quality and reliability concerns, outsourcing opportunities, and expanding complexity, that result from marketplace demands. These programs intend to help organizations develop a higher quality software more effectively and efficiently [6].

Under these circumstances, the SPI framework CMMI has been widely promoted and was already used by over 10000 organizations from more than 100 countries all over the world [7].

Some companies find it a necessity to be CMMI accredited to negotiate and win contracts, others want economic and other benefits as a result. The benefits of implementing this framework include decreased costs, improved delivery schedule, productivity, quality, customer satisfaction, and increased ROI [7, 8].

However, although CMMI has clear benefits, only a small portion of companies adopt it. A study done in companies in Australia [9] shows that small organizations consider CMMI to be infeasible but can see the benefits it would bring, these organizations claim lack of resources and funds required to implement many of the practices. A similar study performed in Malaysia [10] shows that organizations consider the program to be too costly and time consuming, and prefer to focus on other priorities. Furthermore, a study analyzing the situation in China [11] refers the significant amount of resources, including money, manpower, and software, as well as the lack of specialized personnel responsible for quality, having few people with training in SPI programs.

Moreover, a study on SPI success factors [14] states that there is a higher chance of success of implementing CMMI in companies where practitioners are highly motivated for supporting SPI.

Many people reported that adopting CMMI can be quite complicated and often difficult. However, they also acknowledged that the returns from investing in this framework outweigh the expense of implementing it [15, 16].

One reason for the implementation of CMMI to take a lot of time and resources is that CMMI models are complicated. The textual reference models contain a lot of information and organizations need to adopt and integrate multiple practices. There are around two hundred practices in the models with several relationships between them, which makes it difficult to analyze and can be overwhelming for companies.

CMMI models are a set of best practices that focus on what needs to be done to improve performance and not how to do it [17]. For that reason, the existing textual reference models of CMMI-DEV v1.3 [17] and CMMI v2.0 [7] can be ambiguous. The reference models contain very extensive text, various technical concepts and numerous relationships between different practices. Additionally, there are many different concepts in the two most recent versions of CMMI.

With the release of CMMI v2.0, the architecture of CMMI was specifically designed to be flexible, agile, and evolve with the business, technology trends and market demands [7] but it is no less complicated for companies that are already CMMI-DEV v1.3 accredited and need to migrate to CMMI v2.0. If a company is CMMI-DEV v1.3 accredited, they need to migrate to CMMI v2.0 until September 30th 2020 [18], otherwise they will lose the accreditation.

We experienced, first hand, that companies want to delay the migration, as far as it is allow by the CMMI Institute, because of its difficulty.

By using EA models with graphical elements, like ArchiMate, we can provide a less complicated reference model that is more appealing to users and easier to analyze, thus minimizing the impact of a mandatory migration. This model can be helpful for consultants and quality assurance teams, as well as being useful in training sessions.

Therefore, based on the complicated textual reference models, the problem this thesis aims to solve is the difficulty to migrate from CMMI-DEV v1.3 to CMMI v2.0.

3

Theoretical Background

Contents

In this chapter we explain the main concepts necessary to comprehend this thesis. This knowledge is necessary to fully understand the research problem and to formulate an appropriate solution. Hence we present important definitions related to our problem, regarding CMMI, as well as definitions that help to formulate our solution, regarding EA and, more specifically, the ArchiMate modelling language.

3.1 Capability Maturity Model Integration

In 1991, a software development capability maturity model, known as CMM or SW-CMM, was developed by the Software Engineering Institute (SEI), a federally funded research and development center that is part of Carnegie Mellon University. Later, SW-CMM was upgraded to Capability Maturity Model Integration (CMMI) and, in 2010, the model expanded into other areas such as services, acquisition, and people. CMMI is currently managed by the CMMI Institute that, in 2016, was acquired by the Information Systems Audit and Control Association (ISACA) [19, 20].

Originally created for the U.S. Department of Defense to assess the quality and capability of their software contractors, CMMI models have expanded beyond software engineering to help any level of an organization in any industry. This framework expanded its scope to include the entire product life cycle and shifted its focus to organizational development [7, 19, 20].

At its core, CMMI is a proven set of global best practices that enable businesses to improve performance of their key business processes through building, improving and measuring their capabilities. It offers organizations a roadmap for process improvement, organized by critical business capabilities that address the biggest challenges common to any organization [7].

Designed to be understandable, accessible, flexible, and integrate with other methodologies, such as agile, CMMI provides guidelines and recommendations for helping organizations diagnose problems and offers a guide to optimize business results [7]. This framework can be used to measure a company's capability and performance through different appraisal methods. Appraisals assist companies in identifying their processes' strengths and weaknesses, and in examining how much their processes relate to CMMI best practices. The appraisal helps to identify and prioritize their business improvement efforts and eventually obtain a benchmark maturity level (staged representation) or a capability level achievement (continuous representation) [21].

This thesis focus on the most recent version of CMMI, which is CMMI v2.0, and on the migration from CMMI-DEV v1.3. CMMI-DEV v1.3 is one of the Constellations of version 1.3 of CMMI. It contains practices that cover project management, process management, systems engineering, hardware engineering, software engineering, and other supporting processes used in the development and maintenance of products and services [17]. Now, in CMMI v2.0, these constellations were integrated into one single model.

CMMI v2.0 was released on March 2018 and after September 30th 2020 will completely replace the previous version [18].

In this section, we are going to see CMMI v2.0 in more detail, and also some of the differences between CMMI-DEV v1.3 and CMMI v2.0, to fully understand the main concepts.

3.1.1 CMMI v2.0

CMMI v2.0 is an integrated product suite containing five components that, when used together, provide a proven and clear path to meet an organization's business goals. These five components are Training and Certification, Appraisal Method, Model, Adoption Guide, and System Tools [7].

In this thesis, we are focusing on the Model component which provides a path to improvement through evolutionary levels. The CMMI v2.0 Model contains several customized views that can be used in various business settings and enables organizations to achieve their specific goals for performance improvement and organizational success [7].

The structure of the CMMI v2.0 Model is the following [7]:

1. **Categories:** Categories are logical groups or types of views of related Capability Areas that address common problems encountered by businesses when producing or delivering solutions. There are four categories: Doing, Managing, Enabling and Improving;
2. **Capability Areas:** Capability Area is a group of related Practice Areas that can provide improved performance in the skills and activities of an organization or project. There are 10 Capability Areas divided into the 4 categories;
3. **Practice Areas:** Practice Area is a collection of similar Practices that together achieve the defined Intent, Value, and Required Information described in the Practice Area. A Practice Area also includes Explanatory Information containing the practice summary, the related Practice Areas, and the Context Specific Information. There are 25 Practice Areas divided into the 10 Capability Areas;
4. **Practice Group:** Practice Group is the organizing structure for practices within a Practice Area to aid understanding and adoption. It provides a path for performance improvement. Currently, the Practice Groups defined for CMMI are evolutionary levels. There are 5 evolutionary levels (maturity levels);
5. **Practices:** Practices consist of Required Information and Explanatory Information. Required Information is what is essential to understand the full intent and value of the Practice and includes the practice statement, the value statement, and all the additional required information (not every practice has additional required information). The Explanatory Information is the information about the remaining parts of the practice, including Example Activities and Example Work Products, related

Practice Areas (if needed), and Context Specific Information. There are 229 Practices divided into the 25 Practice Areas.

Fig. 3.1 shows the Categories, Capability Areas, and Practice Areas of CMMI v2.0 [7].

Figure 3.1: Categories, Capability Areas and Practice Areas of CMMI v2.0

Additionally, the definition of some of the concepts mentioned in the structure is also important to fully understand the CMMI v2.0 Model.

- **Intent:** Explain what results and accomplishments are expected as an outcome of the Practice

Area;

- **Value:** Business value achievable by adopting the Practices;
- **Additional Required Information:** Further describes the scope and intent, as well as supports clear and consistent understanding and interpretation. May not be present for every Practice Area/Practice;
- **Example Work Products:** Possible outputs of implementing processes that meet a Practice's intent. These serve as guidance and suggestions, not as required work products;
- **Example Activities:** Possible actions that may be taken when implementing processes that meet a Practice's intent. These serve as guidance and suggestions, not as required activities.

In CMMI v2.0, an organization can chose to achieve a **maturity level** or a **capability level**. These levels provide a way to describe their capability and performance, while helping to prioritize certain Practice Areas [7, 21].

Capability levels apply to an organization's process improvement achievement in individual process areas. There are four capability levels, numbered from 0 to 3, and each level builds on the previous levels by adding new functionality or rigor. This means that an organization can be appraised capability level 3 for one Practice Area, but capability level 2 for another [7, 21].

Maturity levels apply to an organization's process improvement achievement in multiple process areas. These levels represent a staged path for an organization's performance and process improvement efforts based on predefined Practices. There are five maturity levels, numbered from 1 to 5, and each maturity level builds on the previous ones by adding new functionality or rigor. The maturity levels cannot be skipped and a particular maturity level is only achieved when all Practices belonging to that level (and all Practices belonging to lower maturity levels) have been successfully implemented [7, 21].

In this thesis, we are focusing on maturity levels. Below is a brief description of each maturity level [7]:

- **Level 0: Incomplete -** Ad hoc and unknown. Work may or may not get completed;
- **Level 1: Initial -** Unpredictable and reactive. Work gets completed but is often delayed and over budget;
- **Level 2: Managed -** Managed on the project level. Projects are planned, performed, measured, and controlled;
- **Level 3: Defined -** Proactive, rather than reactive. The organization-wide standards provide guidance across projects, programs, and portfolios;
- **Level 4: Quantitatively Managed -** Measured and controlled. The organization is data-driven with quantitative performance improvement objectives that are predictable and align to meet the needs of internal and external stakeholders;

- **Level 5: Optimizing -** Stable and flexible. The organization is focused on continuous improvement and is built to pivot and respond to opportunity and change. The organization's stability provides a platform for agility and innovation.

3.1.2 Differences between CMMI-DEV v1.3 and CMMI v2.0

The authors of CMMI v2.0 tried to address issues of older versions of this framework by making a significant amount of changes. These changes include reducing the number of relationships between different CMMI practices, clarifying some ambiguities by changing the terminology, and making CMMI adoption more modular [7, 17].

Therefore, between the two most recent versions of CMMI, new concepts were added, some were changed or replaced, and others were eliminated. Regarding Practices, most are the same but in a different order, some were amplified, and new ones were added. With CMMI v2.0, a new a type of appraisal was introduced, in order to minimize the burden of the appraised organization [22].

Bellow, we describe the most significant changes in terms of structure and terminology [22]:

- A Process Area is now called Practice Area, to emphasize that CMMI is a collection of best practices, not a set of processes to be implemented;
- Practice Areas are broken up into a general descriptive section, called Core Information, and a contextually-applicable section, called Context Specific;
- A Process Area Category is now named Capability Area;
- Required, Expected, and Informative are now referred to as Required and Explanatory. Required content is necessary for achieving an appraisal rating. Explanatory Information is included to aid an accurate interpretation of the Required Information;
- Generic Practices and Generic Goals have been replaced by two institutionalization Practice Areas: Governance (GOV) and Implementation Infrastructure (II), to avoid duplication and improve the intent of institutionalization. Without Generic Practices, the Specific Practices designation is no longer needed;
- Category and Practice Group (Level) are new terms that did not exist in the previous version;
- A Constellation is now called a View, so the three constellations of version 1.3 (Development, Services, and Acquisition) are integrated into a single model.

Regarding the Appraisals, Benchmark appraisals replaces CMMI-DEV v1.3 SCAMPI A appraisals. Evaluation appraisals replaces SCAMPI B and C appraisals. Sustainment appraisals is a new appraisal class which entails a substantially reduced scope to check on process sustainment to ensure maturity is maintained over time.

3.2 Enterprise Architecture

Architecture is the art and science of designing complex structures. EA is the architecture at the level of an entire organization, which provides a holistic view of the enterprise. EA is defined as ”a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure” [19].

In recent years, the popularity of EA increased, having been widely adopted by several organizations. This type of architecture, captures crucial parts of both business and IT, helping to keep the business essentials, while allowing for maximum flexibility and adaptivity. A better alignment between business and IT leads to lower cost, higher quality, better time-to-market, and greater customer satisfaction. Hence, to have a successful business, a good architecture is needed [19, 23, 24].

EA has become an indispensable instrument in controlling the complexity of an enterprise structure, processes, and systems. Through architecture models, views, presentations, and analyzes, the communication gap between architects and stakeholders has been reduced [19, 23] .

In this section, we are going to see in more detail: The Zachman Framework, because it is where the concept of EA for the area of IS first originated, being a baseline for many other EA frameworks; The Open Group Architecture Framework (TOGAF) and the ArchiMate modelling language, because these are the EA standards we use in this thesis.

3.2.1 The Zachman Framework

For the area of IS, the concept of EA originated in The Zachman Framework, published in 1987. The majority of EA frameworks in the industry are methodologies derived from The Zachman Framework, thus it being an ontology for EA [25].

The Zachman Framework provides a structured way for organizations to gain the necessary knowledge about its EA. Zachman proposes a logical structure to categorize and organize descriptive representations of an enterprise in different dimensions and in different perspectives. The foundation of this framework is that an organization does not have just one architecture but many, organized by layers, that allow to represent different perspectives according stakeholder's concerns [26].

It is based in a two-dimensional schema, that results in a matrix. The columns are types of artifacts, that answer the six organization questions (What, How, Where, Who, When, and Why) and the rows are different perspectives, a distinct view from a particular perspective depending on the stakeholder, giving different models for these perspectives (Scope Contexts, Business Concepts, System Logic, Technology Physics and Tool Components) [25].

3.2.2 The Open Group Architecture Framework (TOGAF)

First published in 1995, TOGAF, an Open Group Standard, is a proven EA methodology and architecture framework used by the world's leading organizations [2].

This framework ensures consistent standards, methods, and communication among EA professionals, while addressing crucial business needs. It improves the efficiency of an organization's business and IT operations by assisting the development and maintenance of an EA [2].

The core of TOGAF is the Architecture Development Method (ADM), which provides a step by step approach for developing an EA and helps to establish a framework. It is an iterative process of continuous architecture development and realization, allowing an organization to transform itself to answer business goals and opportunities [2]. The different phases of the ADM are shown in Fig. 3.2.

Figure 3.2: Architecture Development Cycle [2]

Another standard that complements the TOGAF framework is the ArchiMate modelling language, that we talk about in more detail in the next section. ArchiMate viewpoints can be mapped to the different phases of the ADM, as we can see in Fig. 3.3. Combining this modelling language with TOGAF creates a more effective description of an EA [3].

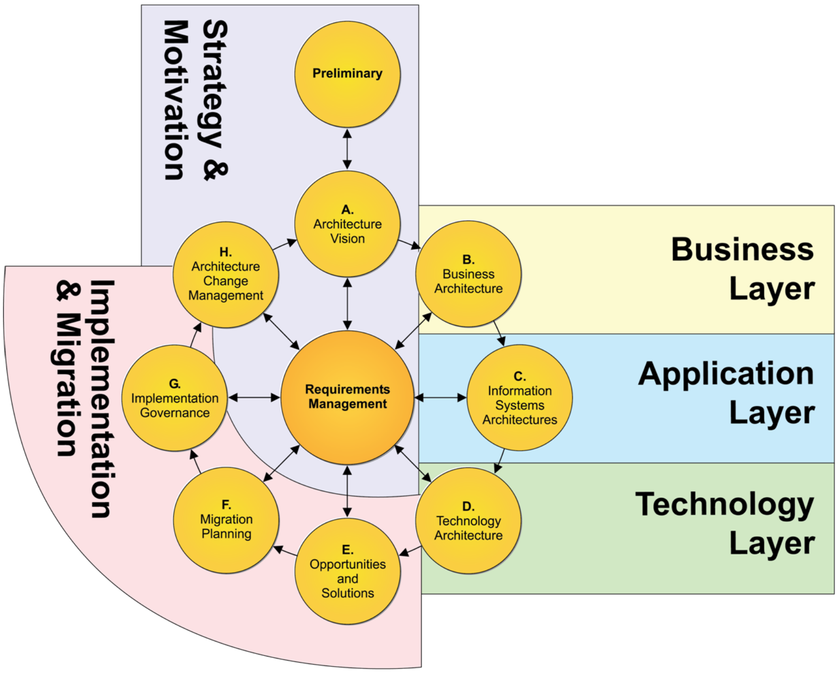


Figure 3.3: Correspondence between the ArchiMate language and the ADM [3]

3.2.3 ArchiMate

ArchiMate, an Open Group Standard, is an open and independent modelling language for EA, that provides a uniform graphical representation for diagrams that describes, analyzes, and communicates many concerns of an EA [4].

This modelling language is universal and strives to be as clear and simple as possible. Its visual aspect makes the communication simple and efficient, improving consistency and comprehensibility [19].

The goal of using ArchiMate is to develop an architecture, and create views of the architecture, that show how to address and balance stakeholders' concerns related with an organization's business and IT systems, as they evolve over time [4, 19].

This language defines generic elements necessary to model an EA. It offers an integrated architectural approach to describe and visualize different architecture domains and their relationships. For this integration to be possible, ArchiMate uses a service-orientation approach to distinguish and relate between different layers and uses realization relationships to relate concrete elements to more abstract elements across these layers [4, 19].

We can distinguish an EA into six layers, being the Business Layer, the Application Layer, and the Technology Layer the main ones [19]:

- **Business Layer:** business services offered to customers, which are realized in the organization by business processes performed by business actors;
- **Application Layer:** application services that support the business, and the application components that realize them.
- **Technology Layer:** technology services needed to run the applications, computer and communication hardware, and system software that realize those services;
- **Strategy Layer:** for modelling the enterprise at a strategic level with its capabilities, resources and the courses of action it may take;
- **Physical Layer:** describes the physical world of equipment, materials, and distribution networks;
- **Implementation and migration Layer:** supports project portfolio management, gap analysis and transition, and migration planning.

Fig. 3.4 shows the several layers in an EA and the aspects. Regarding the aspects, ArchiMate distinguishes, in each layer, several elements in three groups according to its characteristics: passive structure elements, behavior elements, and active elements. The motivation elements model the reasons behind the choices made in the architecture and corresponds to the "Why" in the Zachman framework [19].

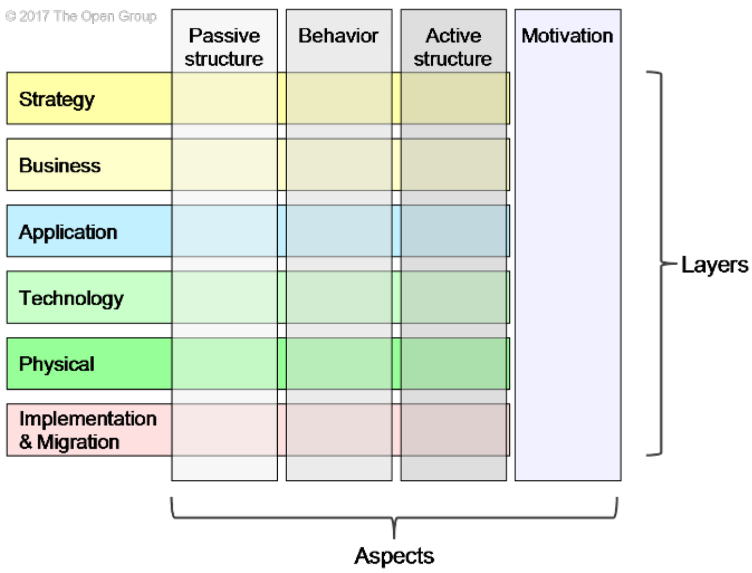


Figure 3.4: Full ArchiMate Framework [4]

3.3 Metamodeling

A metamodel defines the language for expressing a model and provides concepts and relations between concepts necessary for designing any kind of model [27, 28].

Since we are not creating a new modeling language, the first step is to map the concepts of our chosen modeling language, ArchiMate, to the model we want to represent, which is CMMI.

When developing a metamodel, to assure its quality, it is important to follow guidelines. Schutte and Rotthowe proposed a set of guidelines of modelling, divided in the following principles [29, 30]:

- **Principle of Construction Adequacy:** This principle is related to the quality of the model depending on the representation of the reality. This means that the construction of the model should be adequate to its context and purpose, and requires an agreement about the represented problem and type of construction;
- **Principle of Language Adequacy:** This principle describes that the chosen language needs to fit the purpose of the model. The language must be suitable and correct;
- **Principle of Economic Efficiency:** This principle defines that the economic restrictions of modelling information process should be taken into consideration;
- **Principle of Clarity:** This principle is related to the comprehensibility of the model. Hierarchical decomposition, layout design and information filtering help to clarify a model;
- **Principle of Systematic Design:** This principle refers to consistency between different model views, regarding structural and behavioral models;
- **Principle of Comparability:** This principle intents to semantically compare two or more models. This requires an analysis of the models' language and grammar.

A model consists of instances of the object types defined in the metamodel [28]. In this thesis, we are calling the instantiation of our metamodel, reference model. A reference model is an abstract framework or domain-specific ontology which consists on a set of clearly defined concepts that encourage clear communication.

4

Related Work

Contents

This chapter contains the analysis of already existing solutions related to this thesis' context. We perform a literature review of existing researches about subjects related to our thesis, to explain how our proposal differs and adds to the existing ones.

4.1 CMMI-DEV v1.3 Reference Model in ArchiMate

In the master thesis of Valverde et al. [12], the author proposed a graphical reference model for CMMI-DEV v1.3 using ArchiMate as the chosen EA modelling language, to reduce the high perceived complexity of CMMI by its users. The metamodel develop by the author is in Fig. 4.1.

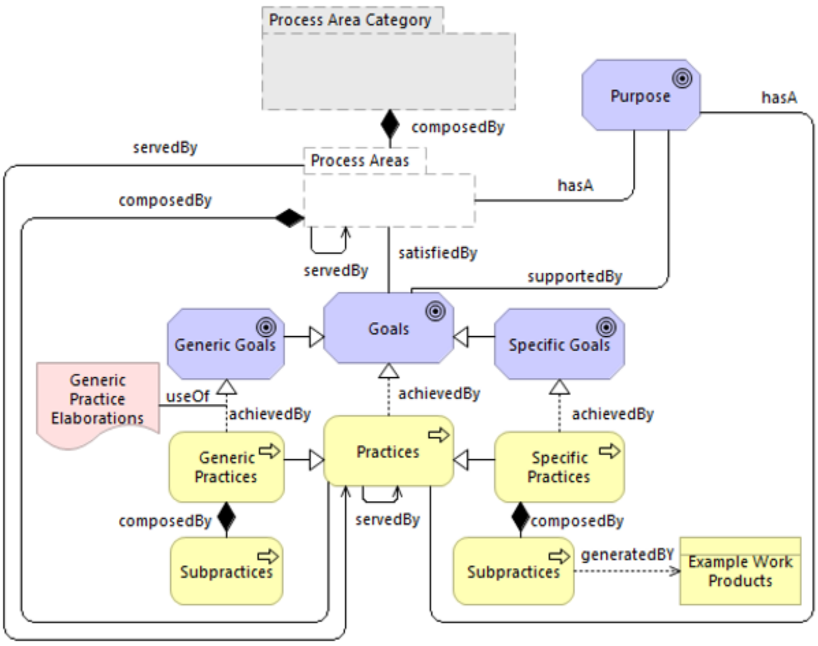


Figure 4.1: CMMI-DEV v1.3 metamodel

This thesis focused on the part of CMMI related to the development of both products and services in version 1.3, called CMMI-DEV v1.3.

To demonstrate the utility of the proposed reference model for CMMI-DEV v1.3, a field study was conducted in a real organization that was improving their processes using CMMI-DEV v1.3. The demonstration in the thesis shows mappings between the EA of the organization and the proposed reference model with the purpose of demonstrating the potential benefits of representing CMMI with an EA.

With the visual representation of the reference model and based on interviews, they believe they were able to lower the user's perceived complexity of CMMI, therefore contributing to turn the CMMI framework easier to use, allowing users to read and understand the CMMI framework more easily and in an interactive way.

This research is a very important contribution for our thesis, since we are doing a follow up from it. We will use the knowledge of this thesis' CMMI-DEV v1.3 reference model for the parts related with CMMI-DEV v1.3 in our thesis.

4.2 CMMI Ontologies

Other than in Valverde et al. thesis [12], there are no researches about CMMI being modelled in Archi-Mate but there are other CMMI ontologies.

Soydan and Kokar [31] proposed a partial formalization of CMMI-DEV. This formalization captures the definitions of a number of concepts of CMMI-DEV and relations among the concepts. The main purpose of the work described in this paper was to demonstrate an automatic determination of a maturity level based upon data of the software engineering processes used by an organization. Towards this aim, a comprehensive formalization of the CMMI-DEV model was expressed in the formal language OWL. Fig. 4.2 presents the second level of detail of this CMMI-DEV ontology.

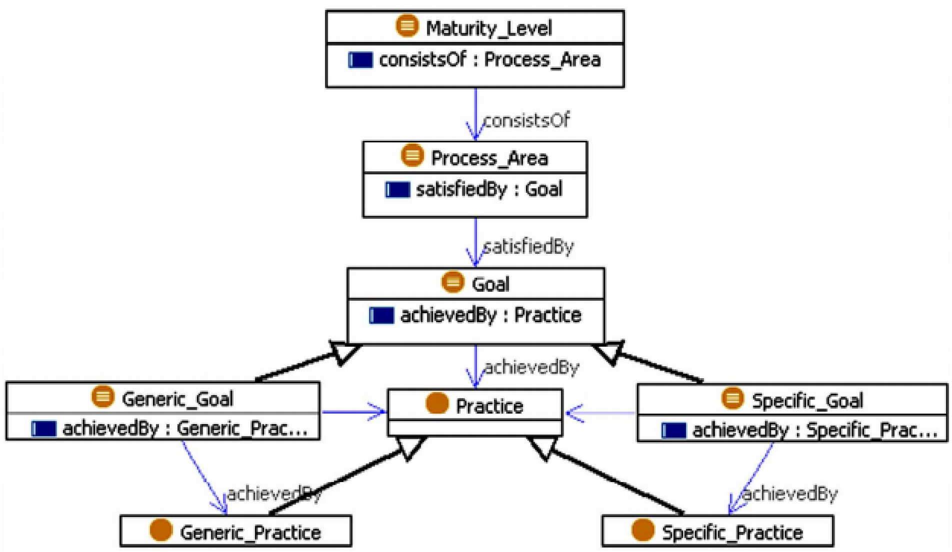


Figure 4.2: Second level of detail of the CMMI-DEV ontology

Musat et al. [32] proposed a Model Driven based tool to automatically generate a language that supports CMMI Process Areas specification. This tool is embedded in a Model Driven Development process and provides a framework that lets the user translate the CMMI generic model into a domain specific model, automatically generating a Domain Specific Language with multiple possibilities of transformation.

4.3 Information Systems Frameworks in ArchiMate

The ArchiMate modelling language was already used to model different IS frameworks. We are going to highlight other researches done using this modelling language.

Teixeira et al. [33] proposed a metamodel of ISO 31000 in ArchiMate. The main objective was to reduce the perceived complexity of ISO 31000, thus facilitating the understanding of the standard. To demonstrate the benefits, the authors applied the blueprints of the standard's main concepts to data from a real company. The proposed ISO 31000 metamodel is in Fig. 4.3.

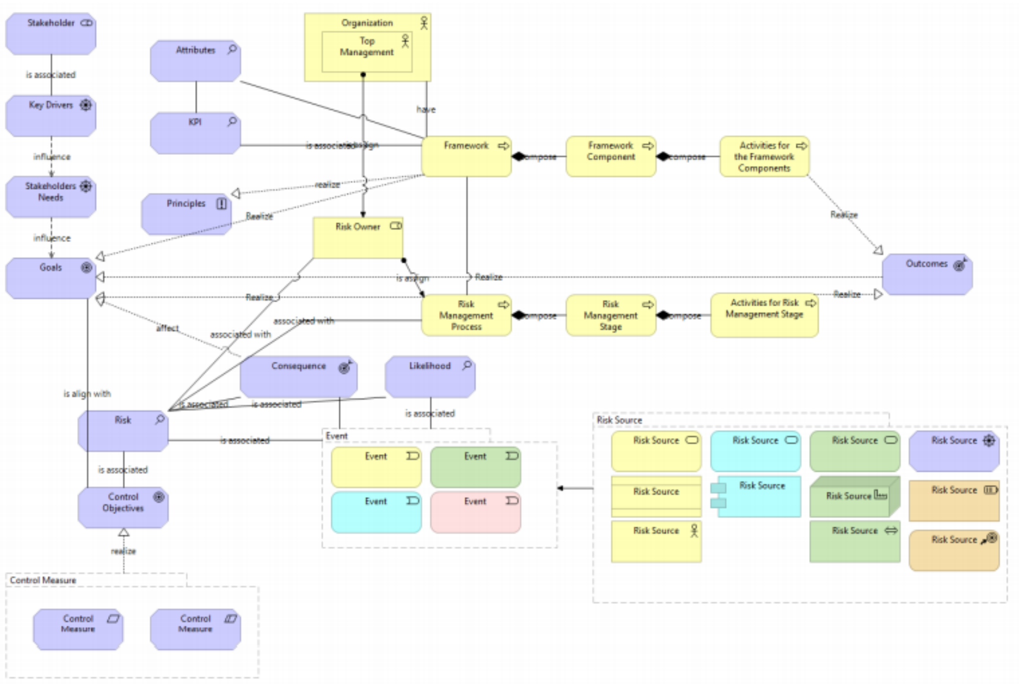


Figure 4.3: ISO 31000 metamodel in ArchiMate

Vicente et al. [34] proposed an Information Technology Infrastructure Library (ITIL) business motivation model in ArchiMate. The goal was to enhance ITIL with a formal representation of its business motivation model. The authors chose ArchiMate's Motivation extension and mapped ITIL motivation to it. The result was a set of consistent models with the whole ITIL motivation.

Silva et al. [35] used ArchiMate to model a process assessment framework. The goal was to enhance Tudor's ITSM Process Assessment (TIPA), a process assessment framework that meets two standards (ISO/IEC 15504 Process Assessment and ITIL), through a EA related notation. The result was a graphical notation of the TIPA Framework using the ArchiMate modelling language, creating a bridge between EA and TIPA, thus making it easier for organizations to improve their processes and achieve desired process maturity levels.

Almeida et al. [36] used ArchiMate to assess Control Objectives for Information and Related Technologies (COBIT) 5 and ITIL implementations. The main goal of this research was to reduce the complexity of mechanisms for IT enterprise governance, by facilitating the assessment of these mechanisms when used simultaneously. To accomplish this, the authors proposed a model in ArchiMate that demonstrates the similarity between the process assessment models for COBIT 5 and ITIL.

Finally, Percheiro et al. [37] proposed a way to represent the ITIL metamodel in ArchiMate, as well as its integration with the COBIT metamodel. The goal was to demonstrate that metamodeling is a useful technique to gain a theoretical foundation and to analyze, compare, and integrate them. To achieve this goal, the authors proposed to model ITIL and associate it with COBIT using ArchiMate.

We can take valuable information from these researches to formulate our proposal. These frameworks have concepts with similar meanings to CMMI concepts, thus we can use the ArchiMate elements the authors used to represent our concepts. Furthermore, these representations help to validate that representing these frameworks with an EA modelling language, like ArchiMate, can reduce complexity and improve communication.

5

Research Proposal

Contents

This chapter corresponds to the phases ”Define the objectives for a solution” and ”Design and development” of DSRM. We present the objectives of our proposed solution to solve the problem identified in chapter 2, followed by a complete description of the proposal.

5.1 Objectives

The main objective is to facilitate the migration for companies that are already CMMI-DEV v1.3 accredited and need to migrate to CMMI v2.0 by providing a reference model in ArchiMate to help with this transition.

To achieve the main objective and, consequently, the solution of our problem, the following objectives need to be accomplished:

- The solution must represent all the main concepts and relationships of CMMI v2.0 and CMMI-DEV v1.3;
- The solution must represent all the main concepts and relationships of CMMI v2.0 in ArchiMate;
- The solution must allow users to navigate to any part of the CMMI v2.0 model.

5.2 Proposal Description

To address the problem identified in chapter 2 and based on previous research work about a CMMI-DEV v1.3 reference model [12], designed in ArchiMate using the modelling tool Archi, we propose a **CMMI v2.0 reference model**, designed in ArchiMate using the modelling tool BiZZdesign Enterprise Studio.

This thesis is a continuation of the research previously done by Valverde et al. [12], regarding a CMMI-DEV v1.3 reference model in ArchiMate. Therefore, we represent CMMI v2.0 also in ArchiMate, so that we have a common thread and can map the differences between the two versions of CMMI.

The chosen language was ArchiMate firstly because of the similarity between CMMI and ArchiMate concepts. ArchiMate allows us to describe and visualise our structure in a clear and simple way, thus anyone related to the business world will be able to understand an ArchiMate model. This language helps us draw a bigger picture, focusing on relationships instead of implementation details. ArchiMate is wider in scope than notations like UML or BPMN, which are domain-specific notations, but is less detailed [4, 19].

This proposal includes the following artifacts:

- Mapping of CMMI v2.0 in ArchiMate and respective metamodel;
- CMMI v2.0 reference model;
- CMMI-DEV v1.3 to CMMI v2.0 visual practice mapping.

5.2.1 Mapping of CMMI v2.0 in ArchiMate and respective Metamodel

To develop a metamodel for CMMI v2.0 using ArchiMate 3.0, we first mapped CMMI v2.0 concepts [7] to ArchiMate concepts [4].

The main concepts of CMMI v2.0 are Categories, Capability Areas, Practice Areas, Practice Groups and Practices.

For the Categories, the Capability Areas and the Practice Groups, we chose ArchiMate's Grouping element (Fig. 5.1). These concepts can be defined as organizing structures, logical groups or types of views [7] and we can use ArchiMate's Grouping element to represent them, since it consists of a composition of concepts that belong together based on some common characteristic [4]. In this case, Categories are composed of Capability Areas, Capability Areas are composed of Practice Areas, and Practice Groups are composed of Practices.

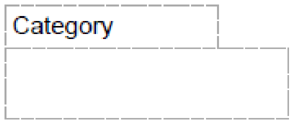


Figure 5.1: ArchiMate's Grouping element notation

To represent the Practice Areas and the Practices, we chose ArchiMate's Capability element (Fig. 5.2). These CMMI concepts pretend to achieve a defined intent and value to the business [7]. This definition conforms to the definition of ArchiMate's Capability element, since it represents an ability that an element possesses and defines what the business does or what it can do. It provides a high-level view of the current and desired abilities of an organization [4], just like CMMI's Practices. We did not use ArchiMate's Business process element to represent these CMMI concepts, since they are not supposed to represent a set of processes to be implemented but to describe the capabilities a company can achieve.

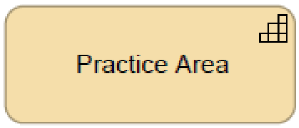


Figure 5.2: ArchiMate's Capability element notation

As said in section 3.1.1, a Practice Area consists of Required Information and Explanatory Information. From these, the main concepts are the Intent, the Value, the Additional Required Information, and the Related Practice Areas.

Similar to a Practice Area, a Practice consists of Required Information and Explanatory Information. From these, the main concepts are the Value, the Additional Required Information, the Example Activities, the Example Work Products, and the Related Practice Areas.

To represent the Intent, we chose ArchiMate's Goal element (Fig. 5.3). The Intent is the explanation of what results and accomplishments are expected as an outcome [7], in other words, it describes what the organization will achieve by satisfying a Practice Area, in turn, ArchiMate's Goal element represents a high-level statement of intent or desired end state for an organization [4].

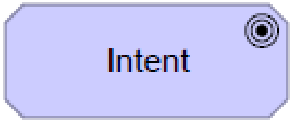


Figure 5.3: ArchiMate's Goal element notation

To represent the Value, we chose ArchiMate's Value element (5.4). CMMI's Value is the business value we achieve by using that component [7] and we can use ArchiMate's Value element to represent it, since it portrays the relative worth, utility, or importance of an element [4].

Figure 5.4: ArchiMate's Value element notation

For the Additional Required Information, we chose ArchiMate's Meaning element (Fig. 5.5). The Additional Required Information is important for clear understanding and interpretation of a Practice Area or Practice meaning [7] and can be represented with ArchiMate's Meaning element, which represents the interpretation of an element of the architecture [4].

Figure 5.5: ArchiMate's Meaning element notation

To represent the Example Work Products we chose ArchiMate's Business object element (Fig. 5.6). Example Work Products are possible outputs of implementing processes that meet the intent of the Practice [7] and can be represented with ArchiMate's Business object, since it could be used to represent information produced and consumed by a business process [4].

To represent the Example Activities we chose ArchiMate's Business function element (Fig. 5.7). Example Activities are possible actions that may be taken when implementing processes that meet the Practice's intent [7] and can be represented with ArchiMate's Business function element, since it is a collection of business behavior based on a set of criteria aligned with an organization [4].

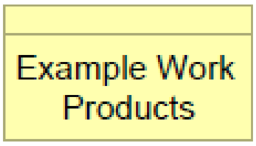


Figure 5.6: ArchiMate's Business object element notation

Figure 5.7: ArchiMate's Business function element notation

The mapping between CMMI v2.0 concepts and ArchiMate concepts is summarized in Table 5.1.

Table 5.1: Mapping of CMMI v2.0 in ArchiMate

CMMI v2.0 concepts	ArchiMate concepts
Category Capability Area Practice Group	Grouping
Practice Area Practice	Capability
Value	Value
Intent	Goal
Additional Required Information	Meaning
Example Work Products	Business object
Example Activities	Business function

In addition to mapping the concepts, it is also important to describe the ArchiMate relationships used in the metamodel.

For the relationships between Categories, Capability Areas, Practice Areas, Practice Groups, and Practices, we chose ArchiMate's Composition relationship (Fig. 5.8). This relationship indicates that an element consists of one or more concepts [4]. In this case, a Category is composed of Capability Areas, each Capability Area is composed of Practice Areas, each Practice Areas is composed of Practice Groups and each Practice Group is composed of Practices.

For the relationship between the Example Activities and the Practices, we chose ArchiMate's Realization relationship (Fig. 5.9), since it represents what needs to be done in the organization in order to realize the Practice [7].

The Related Practice Areas are represented with ArchiMate's Serving relationship (Fig. 5.10). Related Practice Areas represent the Practice Areas that provide something to other Practice Areas [7] and can be represented with the Serving relationship, since it models that an element provides functionality to another element [4].

As for the remaining relationships, they are represented with ArchiMate's Association relationship because those relations cannot be represented by any other relationship.

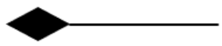


Figure 5.8: ArchiMate's Composition relationship notation

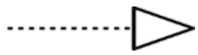


Figure 5.9: ArchiMate's Realization relationship notation

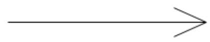


Figure 5.10: ArchiMate's Serving relationship notation

Based on the concepts and relationships that we chose to represent CMMI v2.0 in ArchiMate, and on the guidelines mention in section 3.3, we propose the **CMMI v2.0 metamodel** shown in Fig. 5.11. This representation helps us to get an overview of CMMI v2.0.

5.2.2 CMMI v2.0 Reference Model

The **CMMI v2.0 reference model** is an instantiation of the metamodel presented in the previous section, with the information available in the CMMI Model v2.0 Manual [7].

The full CMMI v2.0 reference model has 4 categories, 10 capability areas, 25 practice areas, a view for each Practice Area and a view for each Practice. In total, our proposed CMMI v2.0 reference model has 255 views. We were granted permission from the CMMI Institute to share part of this information.

This model was presented to CMMI experts and practitioners and corrected according to their feedback, which resulted in two iterations of the reference model.

Additionally, we present some views to facilitate the understanding of CMMI. These additional views can be useful for companies using this framework.

First Iteration

The reference model shows all the main concepts of CMMI v2.0 and their relations. Fig. 5.12 shows part of the model's first level of abstraction. The full image is in Appendix A.

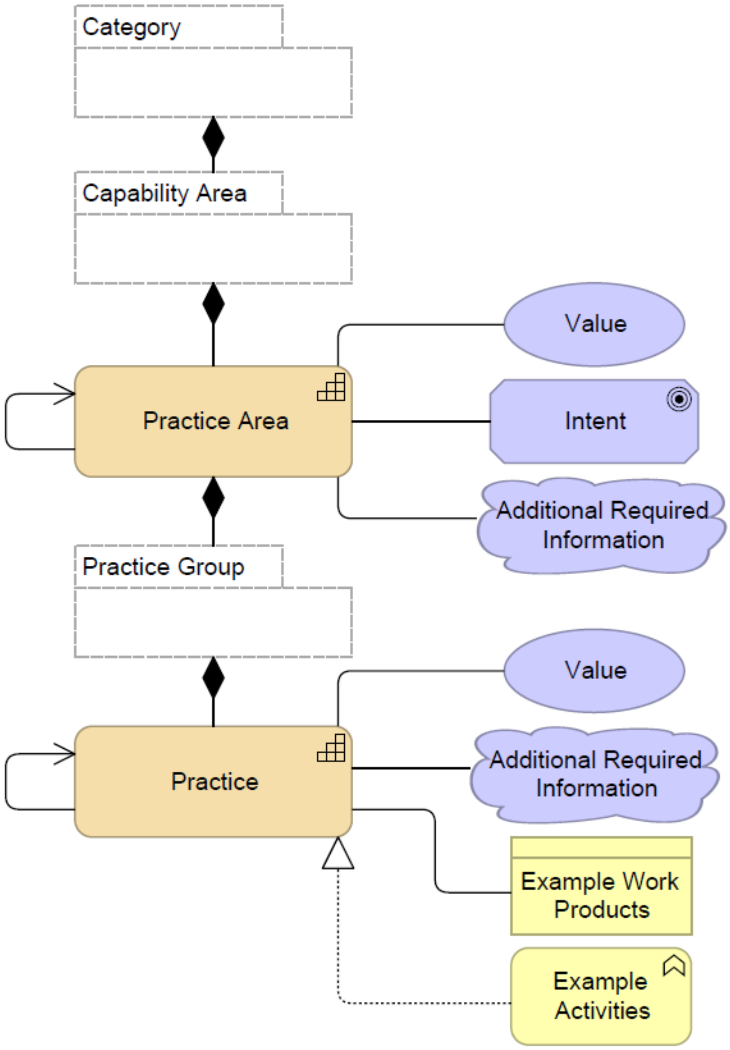


Figure 5.11: CMMI v2.0 Metamodel

As said in section 5.2.1, we used ArchiMate's Grouping element for the Categories and the Capability Areas and ArchiMate's Capability element for the Practice Areas. For the colors of the Categories, we used the colors that CMMI Institute used in Fig. 3.1.

As we can see in this part of the model, the first level of abstraction contains the Category (Doing) and this Category contains four Capability Areas (Ensuring Quality, Delivering and Managing Services, Engineering and Developing Products, and Selecting and Managing Suppliers), each Capability Area contains a set of Practice Areas.

Inside each Practice Area, we have the concept of view. In this case, views help us navigate to other parts of the reference model and see an element in more detail. Therefore, if we double click the view, it will show us a Practice Area individually.

For instance, if we click on "View for VV", we can see the Practice Area Verification and Validation

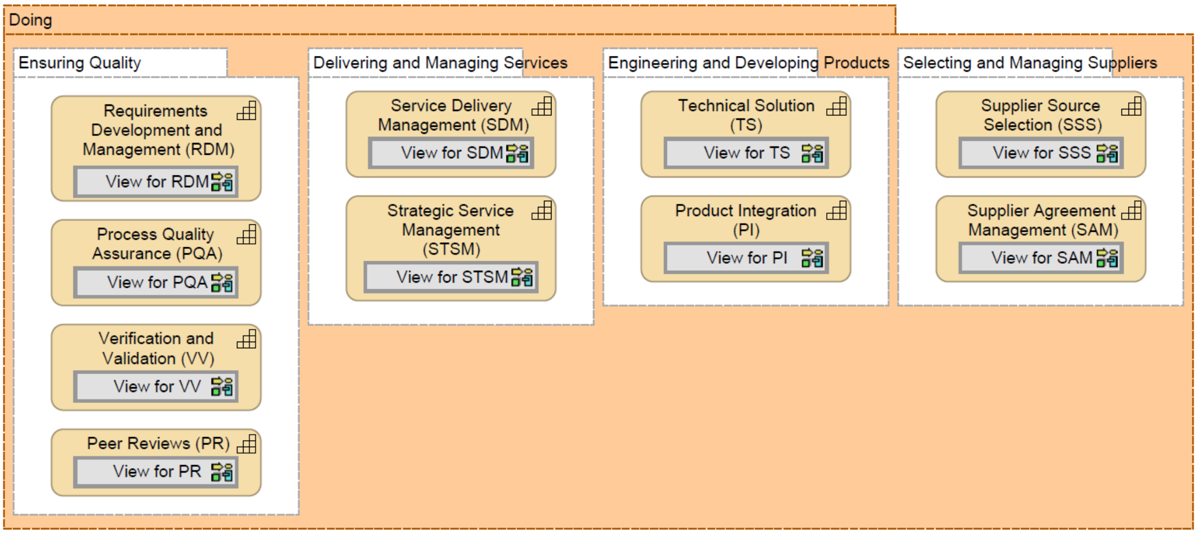


Figure 5.12: Part of CMMI v2.0 Reference model ("Doing" Category)

(VV) in more detail, as presented in Fig. 5.13.

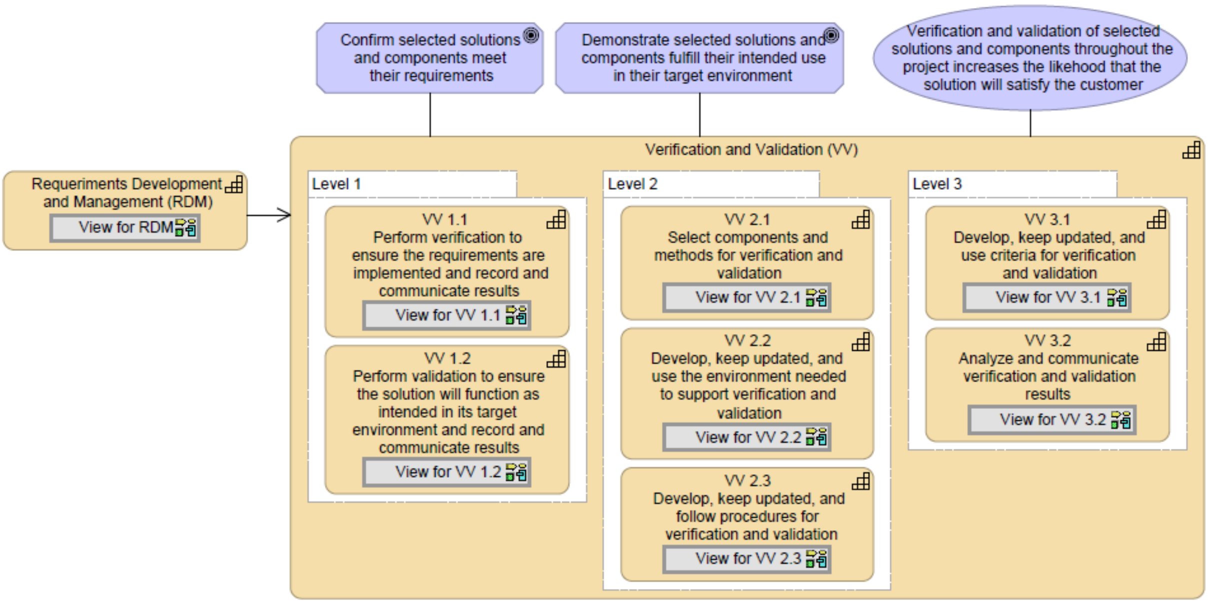


Figure 5.13: Practice Area "Verification and Validation (VV)" - first iteration

In this second level of abstraction of the reference model, we can see: the Practices that compose the Practice Area grouped in levels, which, in this case, go until Level 3; the Value of the Practice Area; the Intent of the Practice Area; and the Practice Areas that contribute to it, called Related Practice Areas. The Additional Required Information element, present in the metamodel (Fig. 5.11), is not represented yet, since in the CMMI Model v2.0 Manual [7] is still blank and will be added in the future.

Inside each Practice, we have again the concept of view, by clicking it, we are able to analyze each Practice individually and see its Value, Example Work Products, Example Activities, and Related Practice Areas. Due to this information not being public, CMMI Institute does not allow us to share it.

This version of the reference model was shared with CMMI experts and practitioners to collect feedback regarding the correctness and utility of our model. As a result, changes were made to the second level of abstraction of the reference model (Fig. 5.13).

Second Iteration

The majority of suggestions made by CMMI experts and practitioners regarded the Value and Intent concepts. These elements are mandatory in CMMI v2.0, thus needing a bigger focus and distinction.

Therefrom, instead of the Value and Intent being represented on top of the Practice Area, connected using ArchiMate's association relationship, they are now inside of the Practice Area and the name Value and Intent is written on them. These changes help to do a better distinction between these concepts and make them look part of the model. Also, the concept of them being mandatory was introduced by having a box saying exactly that. The final version is in Fig. 5.14.

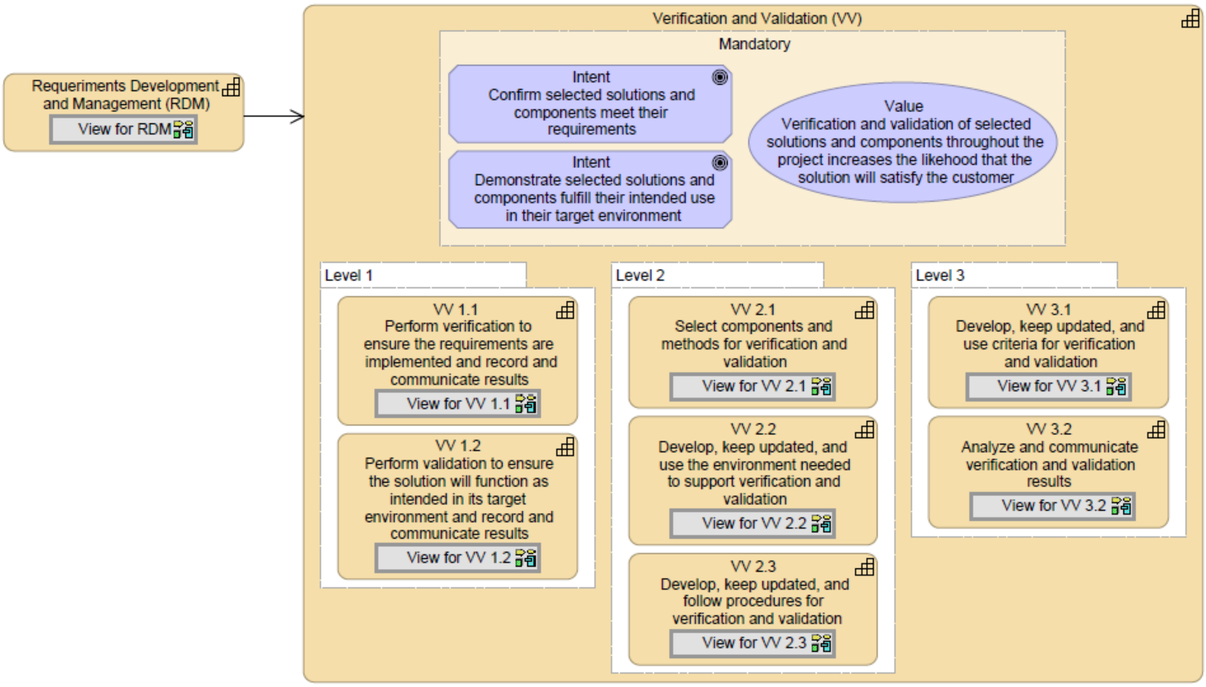


Figure 5.14: Practice Area "Verification and Validation (VV)" - second iteration

Additional Views

Additionally to the views described in the CMMI Model v2.0 Manual [7], we propose two more views of the reference model to convey the information for a specific purpose. The focus of these views is not to achieve a maturity level, it is for companies to see what value implementing certain Practices will bring them.

The Value and the Intent of a Practice Area are very important for a company. These descriptions of what business value it brings and the results and accomplishments expected as an outcome, can be a decisive factor in whether or not to adopt a Practice. Considering this, we provide a view that collects the Intents and Values for the Practice Areas in each Capability Area. Fig. 5.15 shows the Intents and Values for the Practice Areas in the "Supporting Implementation" Capability Area.

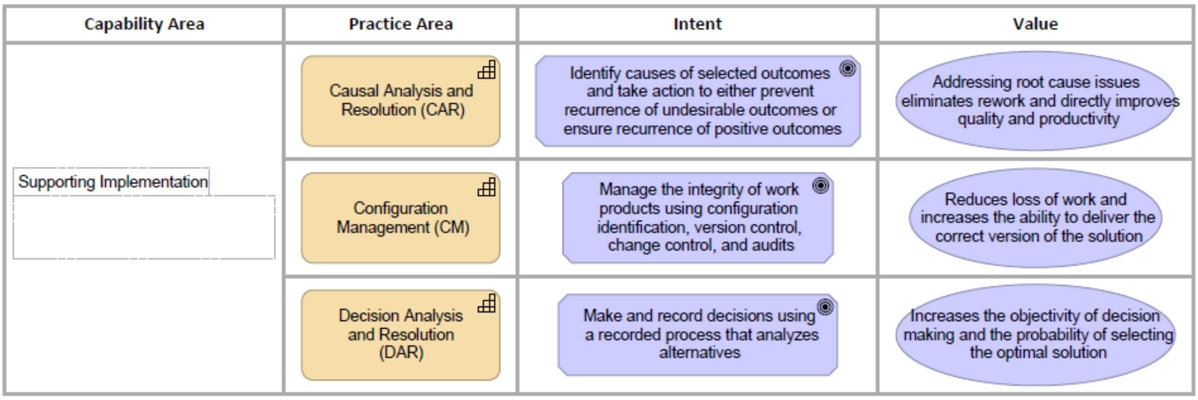


Figure 5.15: Intents and Values for the Practice Areas in the "Supporting Implementation" Capability Area

Also based on the importance of the Value for companies, we grouped Practices by their Value type. We considered Value type as being the type of business value companies want to achieve. Some examples of Value types are "Reduce", "Increase", "Avoid", and "Maintain".

This view, presenting the Practice's Values grouped by Value type, can be useful for companies to choose what Practices to implement, based on the business value they want to achieve. Part of this view for the Value type "Maintain" is shown in Fig. 5.16.

In Fig. 5.16, the first column has the Value type, the second column has the Practices and the third column has the full description of the Value that implementing those Practices would bring.

5.2.3 CMMI-DEV v1.3 to CMMI v2.0 visual practice mapping

The main objective of this proposal is to facilitate the migration for companies that are already CMMI-DEV v1.3 accredited and want to migrate to CMMI v2.0. Therefore, a visual mapping between CMMI-DEV v1.3 and CMMI v2.0 practices is useful.

CMMI Institute provided an Excel file that contained CMMI-DEV v1.3 to CMMI v2.0 practice mapping. Even though according to the Excel file CMMI-DEV v1.3 practices have direct correspondence to CMMI v2.0 practices, for some of the practices that correspondence is not complete. Thus, we analyzed the differences between the equivalent practices. Thereby, following this Excel, we identified the

Value type	Practices	Value
Maintain	CONT 2.2 Identify and prioritize resources essential for continuity	Maintain customer satisfaction and continue business operation during an emergency or significant disruption
	IRP 2.1 Develop, keep updated, and follow an approach for incident resolution and prevention	Maintain customer satisfaction by addressing incidents in a consistent and efficient manner
	SAM 2.4 Manage invoices submited by the supplier according to the supplier agreements	Maintains a good business relationship between the acquirer and supplier
	SAM 1.3 Process supplier invoices	Maintains a good working relationship with suppliers while meeting agreements
	SSS 2.1 Develop a solicitation package and keep it updated	Maintains the integrity of the solicitation package for the objective comparison and evaluation of proposals

Figure 5.16: Part of the view for the Value type "Maintain"

changes that occurred in terms of Practice Areas and identified the gaps between CMMI-DEV v1.3 and CMMI v2.0 practices. For instance, the Verification (VER) Process Area of CMMI-DEV v1.3 does not exist alone in CMMI v2.0. These Practices now equate to Practices in the Practice Areas VV and Peer Reviews (PR) of CMMI v2.0. Also, the Validation (VAL) Process Area of CMMI-DEV v1.3 does not exist alone in CMMI v2.0. These Practices now equate to Practices in the VV Practice Area of CMMI v2.0. Thereby, these two Process Areas of CMMI-DEV v1.3 merged into one new Practice Area of CMMI v2.0, VV, and the Practice Area PR was created.

Furthermore, even though most of the Practices have a direct correspondence between the two versions of CMMI, from VER and VAL to VV Practices, in CMMI v2.0 it was added to communicate the results of performing validation and verification activities, as well as communicate the results of analyzing those activities. Thus, the correspondence between some of the equivalent Practices is not complete. Also, from VER to PR, it was introduced in CMMI v2.0 to keep the procedures and materials updated, resolve the issues encountered during peer analysis and analyze the peer analysis results.

On the top of Fig. 5.17, we can see the VAL Specific Practices of CMMI-DEV v1.3 and, under it, the VV Practice Area of CMMI v2.0. The equivalent Practices are represented using circles of the same color. If a correspondence is not total, it is represented with a dashed circle.

The circles' colors were chosen based on the same Practice type being highlighted in the same tone. Therefore, practices of the type "Select" are in yellow tones, "Establish" are in orange tones, "Perform" are in green tones, "Prepare" are in pink tones, "Conduct" are in grey tones, and "Analyze" are in blue tones.

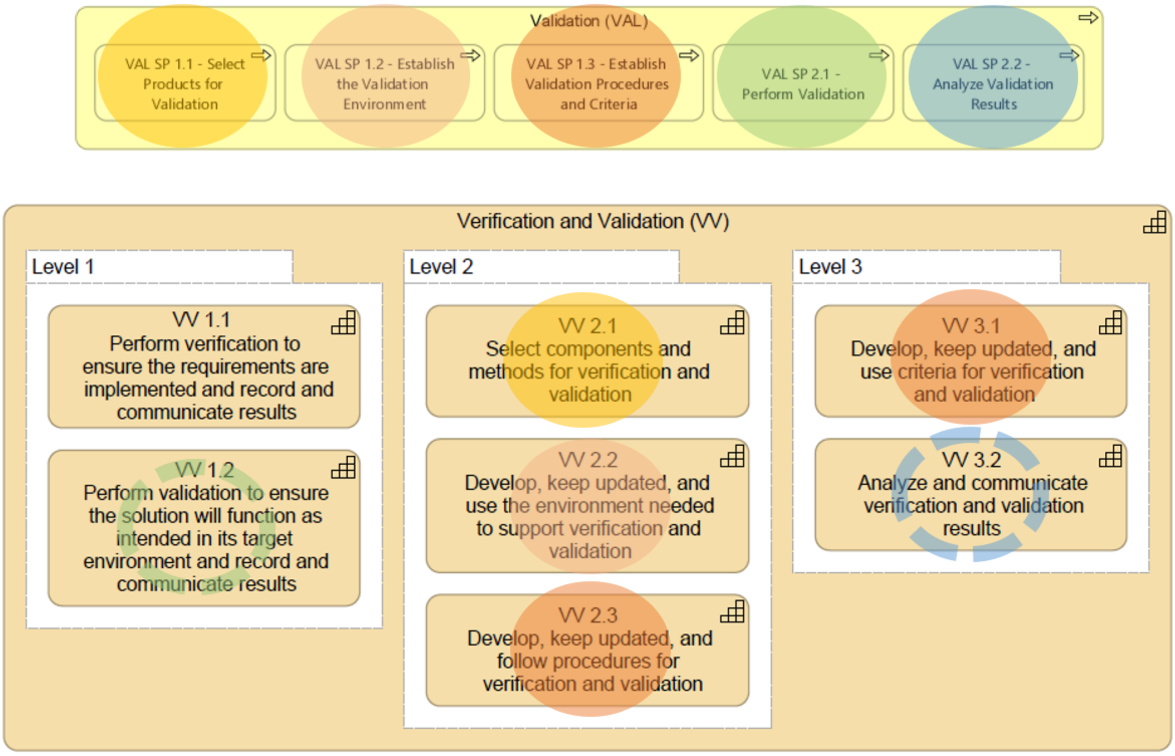


Figure 5.17: Mapping between VAL (CMMI-DEV v1.3) and VV (CMMI v2.0)

6

Demonstration

Contents

This chapter corresponds to the DSRM's demonstration phase. The purpose of this demonstration is to prove the utility of the proposed solution, explained in section 5.2, by demonstrating that our proposal can solve the problem described in chapter 2.

To demonstrate that the solution we developed solves the identified problem and achieves the defined objectives, we applied it on a real-world organization. This way, we were able to validate our proposal with the relevant stakeholders.

We chose a Portuguese IT company specialized in the development of banking software. This company is CMMI-DEV v1.3 Maturity Level 3 accredited and, until September 30th 2020 [18], needs to migrate to CMMI v2.0, to continue being CMMI accredited.

We started working with the company in November 2018 and presented them the final version in September 2019.

In Fig. 6.1, we have the scope of Valverde et al. thesis [12] in the blue box and the scope of this thesis in the green box. For the demonstration, Valverde et al. [12] modelled the company's AS-IS EA, mapped it to CMMI-DEV v1.3, and then modelled the TO-BE EA according to CMMI-DEV v1.3 Practices.

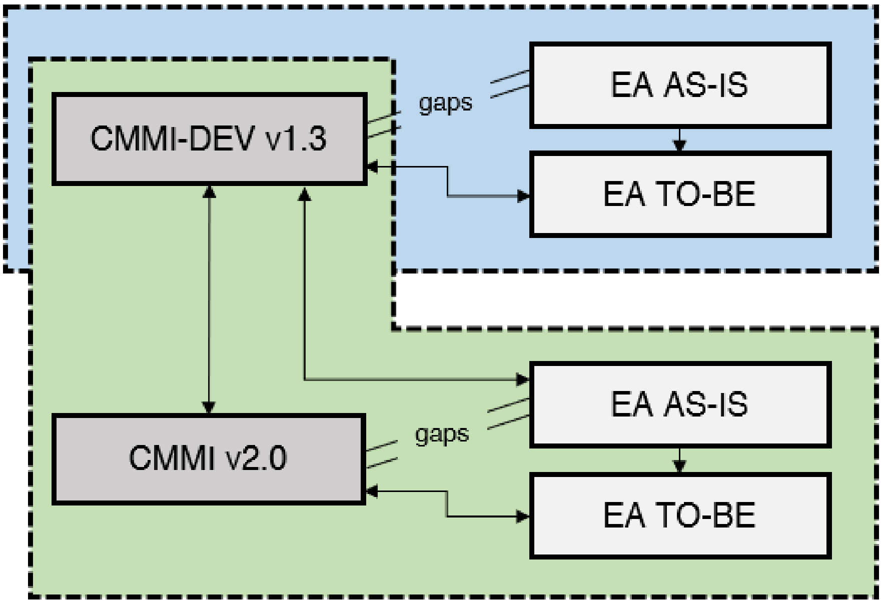


Figure 6.1: Thesis Scope

Our demonstration was done using the ArchiMate modelling language and the BiZZdesign Enterprise Studio modelling tool, and comprises the following steps:

1. Map company's procedures to CMMI-DEV v1.3;
2. Map company's procedures to CMMI v2.0;
3. Model the transition.

Firstly, the company chose two process areas from CMMI-DEV v1.3 that, for them, were the most relevant. The chosen process areas were **VER** and **VAL**.

The models and mappings were frequently shown to a person involved in the company's CMMI accreditation. Our solution was refined and continuously improved according to their feedback.

6.1 Mapping between company's procedures and CMMI-DEV v1.3

As proposed by Valverde et al. [12], the first step is to model the AS-IS of the organization's EA. To do so, we read the documentation provided by the company about its procedures.

After analyzing the company's documentation, we concluded that the procedures that answered Practices from the VER Process Area of CMMI-DEV v1.3 consist of four small procedures from the company's Software Development process. These procedures are: Certification Tests, Integration Tests, Unit Tests, and Peer Reviews. The procedure that answered Practices from the VAL Process Area of CMMI-DEV v1.3 is the Certification Tests procedure, also from the Software Development process.

Therefrom, we modelled, in ArchiMate, these four procedures. Fig. 6.2 shows the Certification Tests procedure.

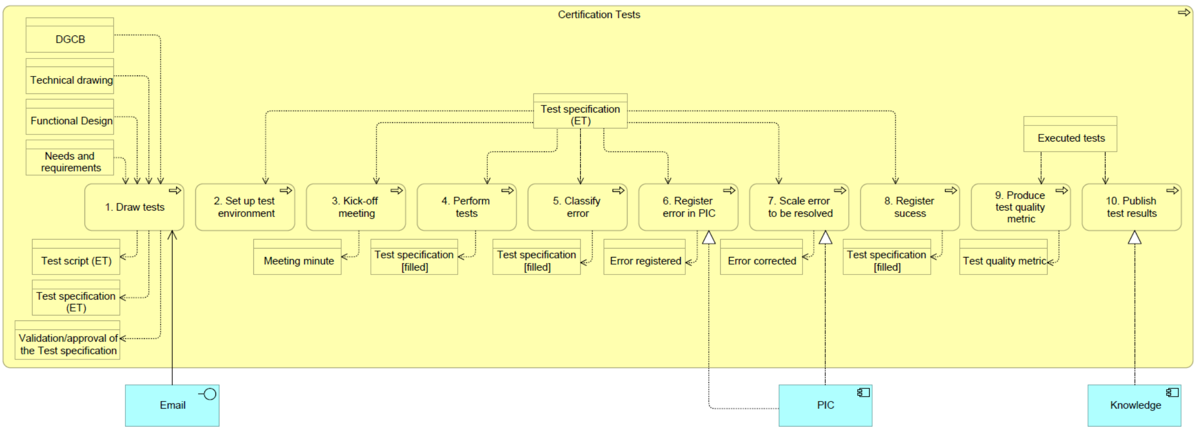


Figure 6.2: Company's Certification Tests procedure

For the mapping with CMMI, the only relevant information are the actions that are performed and the inputs and outputs of those actions, thus we only modelled these elements. Additionally, we added the tools used to perform those actions because the company was in the process of changing them and it would be interesting to see the tools they are using now versus the tools they will use in the future.

We used ArchiMate's business process concept for the actions and, for the inputs and outputs of those actions, we used ArchiMate's business object concept. For the tools, we used ArchiMate's application interface and application component concepts.

The second step was to map the company's procedures to CMMI-DEV v1.3. Therefore, we identified which actions satisfied the VER and VAL Specific Practices and represented it using circles of the same color. If a circle is divided in more than one color, it means that that action satisfies more than one Practice.

On the top of Fig. 6.3, we can see the VAL Specific Practices of CMMI-DEV v1.3 and, under it, the company's procedure that implements those Practices, which is the Certification Tests procedure. The mappings for the procedures that implement VER Practices are in Appendix B.

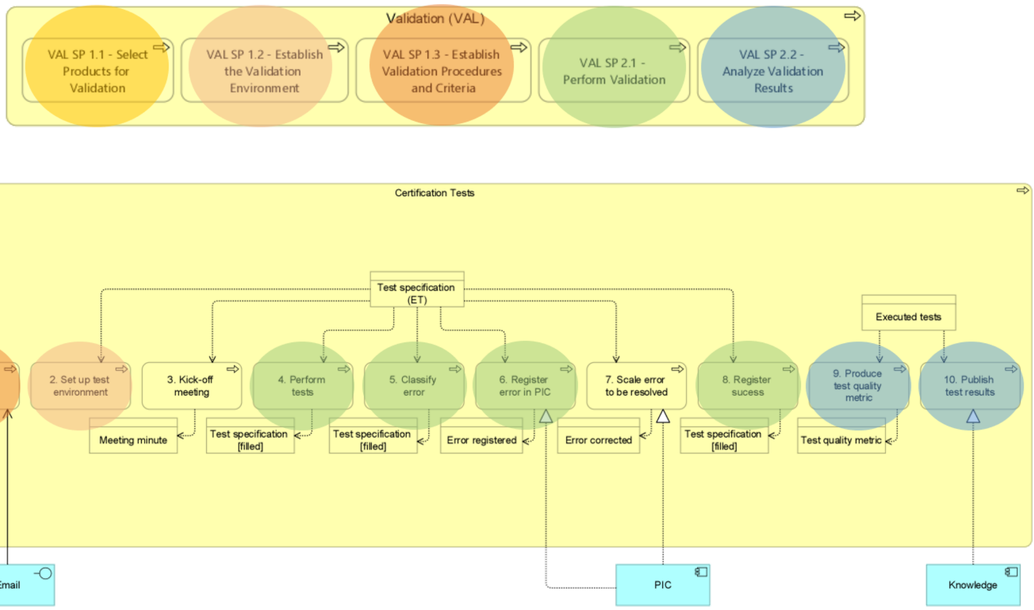


Figure 6.3: Mapping between VAL Specific Practices and Certification Tests procedure

Since the company was already CMMI-DEV v1.3 Maturity Level 3 accredited, their procedures answered all the CMMI-DEV v1.3 VER and VAL Practices.

6.2 Mapping between company's procedures and CMMI v2.0

The second phase consisted of mapping the company's procedures to CMMI v2.0 and identifying what Practices of CMMI v2.0 are not being answered in the company's current state (AS-IS).

This kind of mappings presents the gap analysis in a different way. We can identify what Practices of CMMI v2.0 are not being satisfied in the company's current state through the dashed circles or a missing color in the procedures. Thereby, the company knows in what they should focus when changing their EA to be compliant with CMMI v2.0.

On the top of Fig. 6.4, we can see the PR Practice Area of CMMI v2.0 and the respective Practices.

Under it, we have the company's Peer Reviews procedure, that is the procedure in which these Practices should be answered.

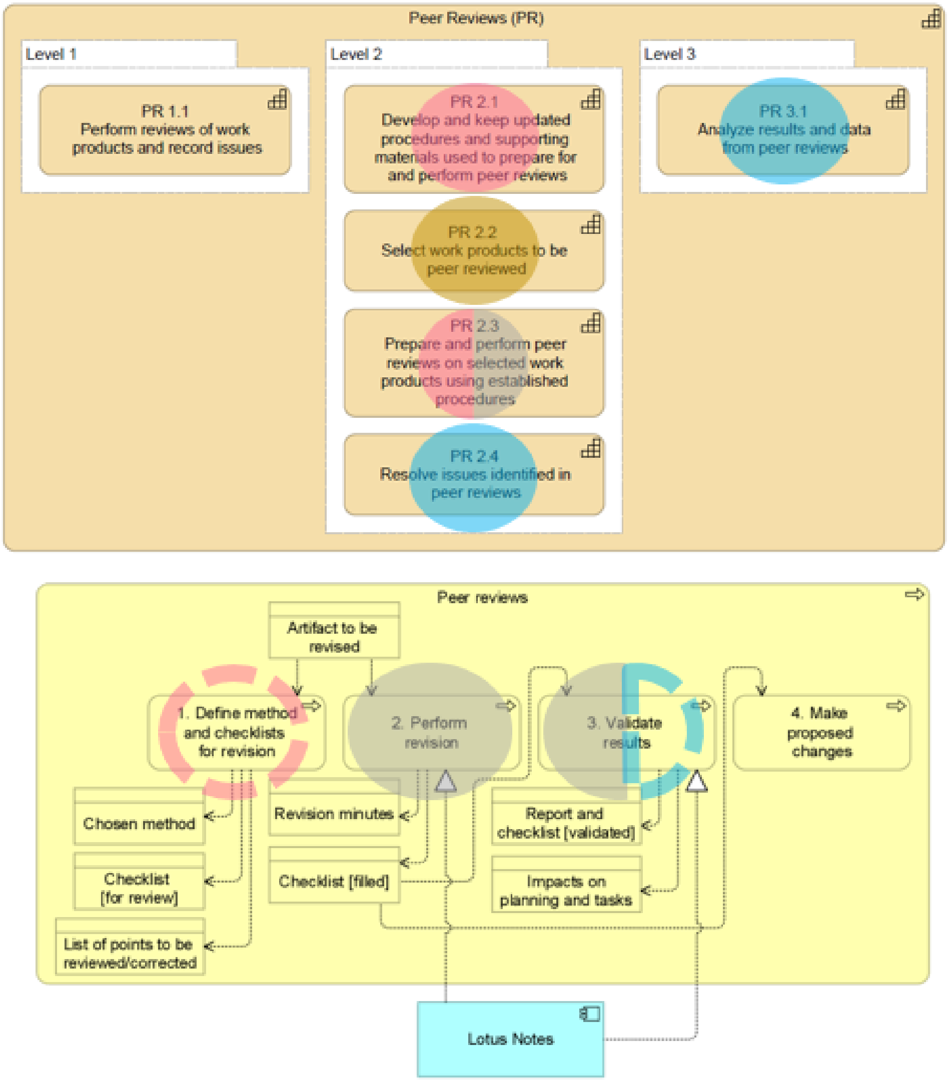


Figure 6.4: Mapping between PR Practices and Peer Reviews procedure

As mention in section 5.2.3, the PR Practice Area did not exist alone in CMMI-DEV v1.3, it was separated from the VER Process Area of CMMI-DEV v1.3 and given a bigger focus.

As we can see in Fig. 6.4, the company's current EA does not fully satisfy the Practices "PR 2.1" and "PR 2.3" and does not answer the Practice "PR 2.2". Therefore, the company needs to make changes in their procedures to fully satisfy these CMMI v2.0 Practices.

The mappings for the VV Practice Area of CMMI v2.0 are in Appendix C.

6.3 Modelling the transition

The third and final phase of this demonstration was to model the TO-BE of the company's procedures compliant with CMMI v2.0.

In the previous section, we were able to identify the gaps between the two versions of CMMI. Therefrom, we modelled the desired state of the company's EA fully satisfying the VV and PR Practice Areas of CMMI v2.0.

Since this was a migration from CMMI-DEV v1.3, it was important to see the transition from the current state of the procedures to the desired state. Therefore, we chose ArchiMate's Migration viewpoint and we can see the AS-IS procedures, as well as the TO-BE procedures.

ArchiMate's Migration viewpoint is used to model the transition from an existing architecture to a target architecture. We used the plateau and gap concepts from this viewpoint. A plateau represents the state of the architecture in different stages in time, having a baseline and a target architecture that describe the current state and the desired future state, and, between those, we have transition architectures. A gap represents the differences between two plateaus and is an important outcome of the gap analysis thus being important to plan the migration [4].

In Fig. 6.5, we have this representation for the company's Peer Reviews procedure.

Our baseline was the company's EA compliant with CMMI-DEV v1.3 and our target was the company's EA compliant with CMMI v2.0. The transition was to identify the gaps, which we did in the previous section. The gap between our baseline architecture and our transition architecture was obtained by mapping the company's procedures with CMMI v2.0. The gap between our transition architecture and our target architecture was the processes and outputs needed for the procedure to be compliant with CMMI v2.0.

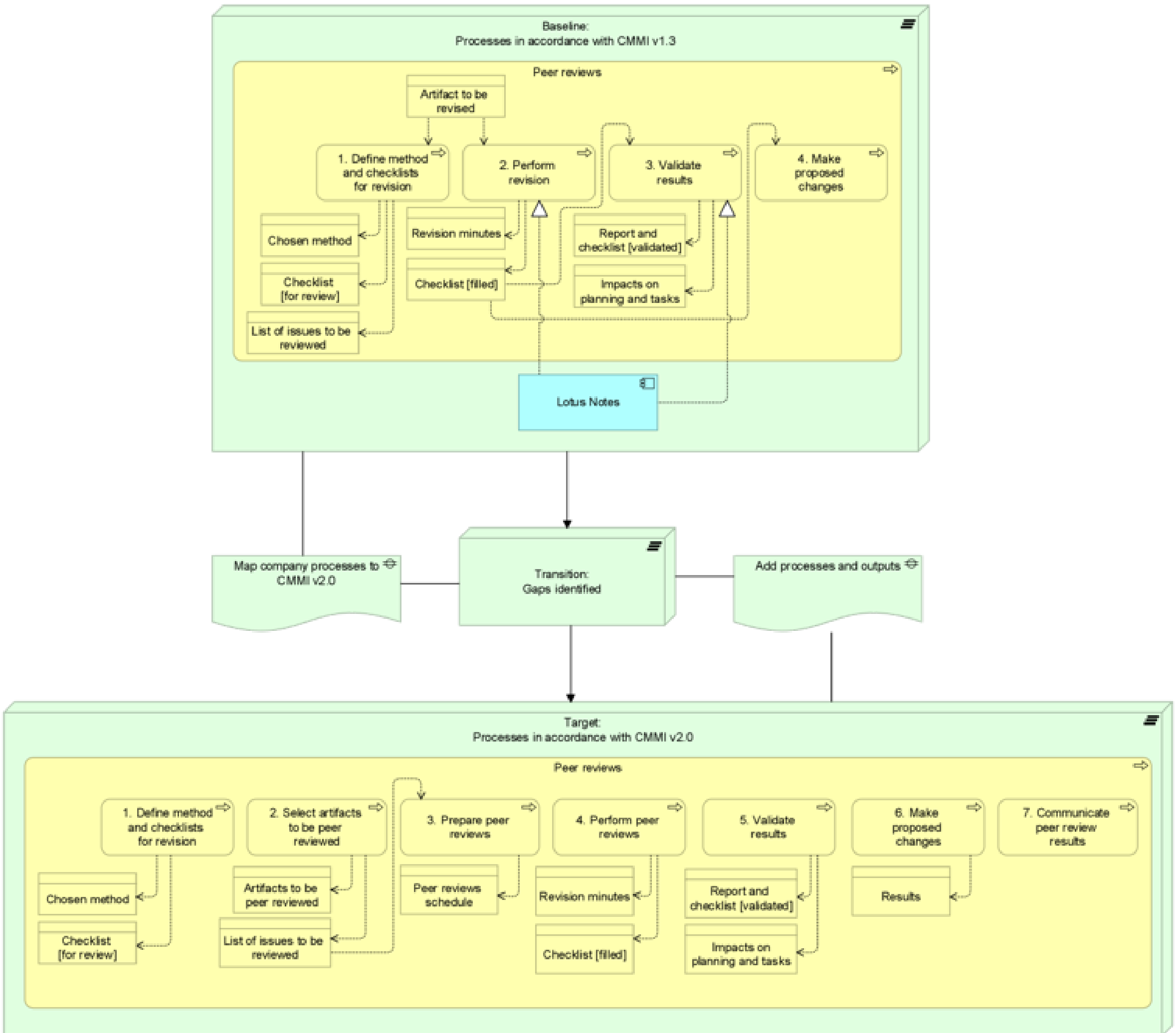


Figure 6.5: Migration viewpoint of the Peer Reviews procedure

7

Evaluation

Contents

This chapter corresponds to the DSRM's evaluation phase. Here, we pretend to evaluate how well our proposed artifacts represent a solution to the problem identified in chapter 2.

Therefore, to validate if our proposed solution is effectively achieving the defined objectives, is useful to users and easy to use, we used the following methods:

- **Wand and Weber method** to evaluate the mappings;
- **Moody and Shanks quality management framework** to evaluate the quality of our model artifact;
- **Questionnaire** to collect quantitative data and feedback about our proposal from CMMI experts and practitioners;
- **Interview** with people involved in the demonstration to evaluate our proposal through its application.

7.1 Wand and Weber method

Wand and Weber method allows us to compare two grammars and examine their ontological completeness and ontological clarity. We used this method to analyze the concept mapping between CMMI v2.0 and ArchiMate (section 5.2.1) by identifying the following ontological deficiencies [5]:

- **Incompleteness:** can each concept from the first set be mapped with a concept from the second set? (the mapping is partial if it is not total);
- **Redundancy:** can each first set concept be mapped on more than a single second set concept? (the mapping is redundant if it is ambiguous);
- **Overload:** can each second set concept be mapped with exactly one or more than one first set concept? (the mapping is overloaded if at least one concept of the second set is mapped to a concept of the first set);
- **Excess:** can each second set concept be mapped on a first set concept (the mapping is excessive if there are second set concepts without a mapping).

The ontological deficiencies regarding the concept mapping between CMMI v2.0 and ArchiMate are illustrated in Fig. 7.1. We considered the first set to be CMMI concepts and the second set to be ArchiMate concepts.

Regarding **incompleteness**, for each CMMI concept (first column of Table 5.1) there is a correspondence to an ArchiMate concept (second column of Table 5.1). Therefore, our mapping is complete.

As for **redundancy**, CMMI concepts can only be represented by one ArchiMate concept. This may be due to CMMI concepts being well defined and described, thus not being ambiguous. Therefore, our mapping is not redundant.

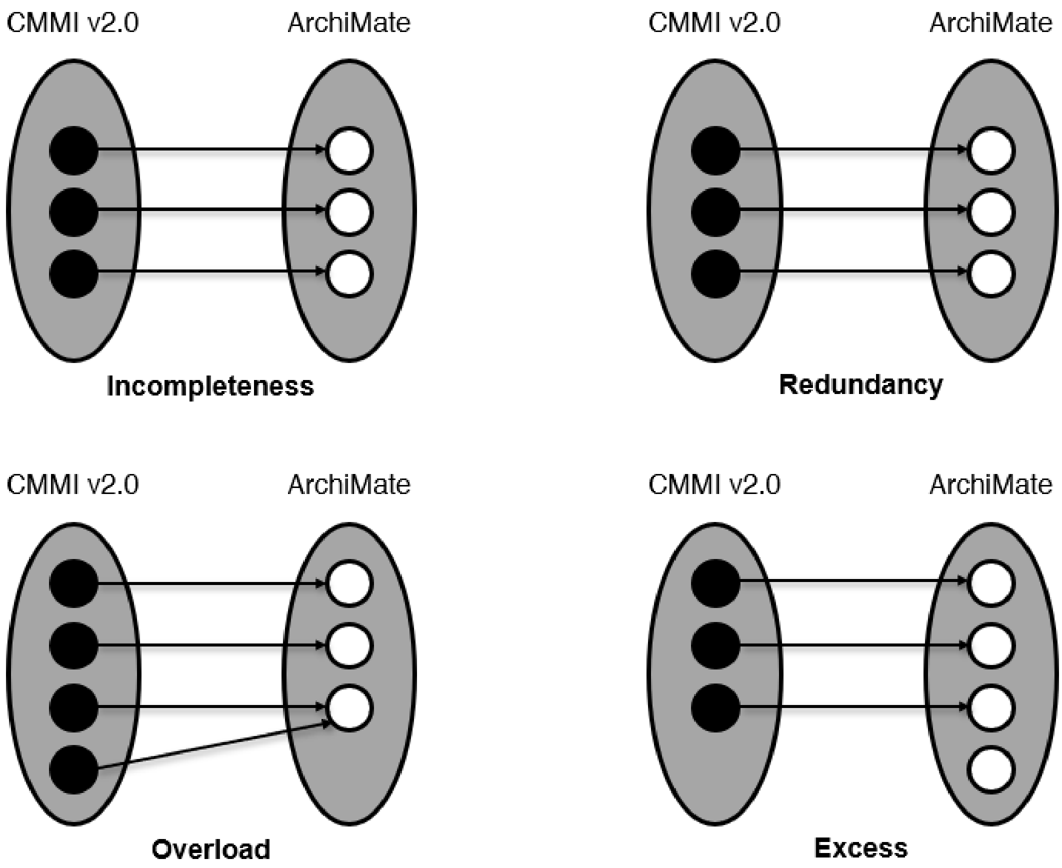


Figure 7.1: Ontological deficiencies (adapted from [5])

We found **overload** because there are several CMMI concepts to only one ArchiMate concept. For instance, ArchiMate's grouping element is used to represent Categories, Capability Areas, and Practice Groups. As long as we do not do the reverse process and go from the CMMI v2.0 reference model in ArchiMate back to the CMMI v2.0 textual reference model, none of the overloaded concepts will cause problems.

Lastly, we found **excess**. ArchiMate comprises a much larger scope than what we used to represent CMMI. Therefore, is normal that it has many excess concepts that are not defined in CMMI, for instance, concepts from the application and technology layers of ArchiMate. This excess can cause problems when attempting to derive a CMMI model from an ArchiMate model, as in CMMI there is nothing to map these excess concepts with.

To conclude, we found instances of two deficiencies, overload and excess. However, they do not represent a major problem while modelling. The excess does not present any issue and the overload can be fixed by adding a property to the ArchiMate elements which allows to distinguish between the different CMMI concepts.

7.2 Moody and Shanks quality model framework

The Moody and Shanks quality model framework allows us to evaluate and improve the quality of data models. We used this framework to assess the quality of our model artifact regarding the following quality factors [38]:

- **Completeness:** refers to whether the data model contains all user requirements;
- **Simplicity:** means that the data model contains the minimum possible entities and relationships;
- **Flexibility:** is defined as the ease with which the data model can cope with business and/or regulatory change;
- **Understandability:** is defined as the ease with which the concepts and structures in the data model can be understood;
- **Correctness:** is defined as whether the model conforms to the rules of the data modelling technique;
- **Implementability:** is defined as the ease with which the data model can be implemented within the time, budget, and technology constraints of the project.

Regarding **completeness**, our metamodel includes all the relevant concepts and relationships described in CMMI v2.0, thus being complete.

As for **simplicity**, CMMI contains many concepts but we only modelled the main ones, which are the ones containing relevant information to build our reference model. Also, with the use of views, we were able to have different levels of abstraction, each level containing the minimum possible concepts and relationships, avoiding the CMMI v2.0 reference model to became one large complex model.

Concerning **flexibility**, our metamodel is aligned with CMMI which is designed to be flexible and adapt to any type of company or project. Business changes would not affect our model, only changes in the CMMI model. However, our model structure is simple, if a change is needed, it would be easy to adjust, thus we can consider our model to be flexible.

Regarding **understandability**, the concepts and structures in our metamodel are related with CMMI and ArchiMate, which makes it easy to recognize for people in these areas. In the interview, after a brief explanation, the interviewees were able to understand our proposal and said that even someone with no knowledge in any of those areas would be able to understand our model. Also, both in the interview and in the questionnaire's answers, it was mentioned that this would be useful for someone that is learning CMMI, thus we can conclude that our model is easy to understand (section 7.3 and 7.4).

As far as **correctness** is concerned, our metamodel was constructed respecting CMMI and ArchiMate specification. By mapping CMMI concepts to ArchiMate concepts that better represented them and using ArchiMate representation and colors, our model conforms to the rules of the EA modelling language. Additionally, the questionnaire's results show that this is a good way to represent the CMMI framework (section 7.3).

Finally, for **implementability** our metamodel is simple and aligned with CMMI, thus being aligned with the organization's processes. The implementability of our model was demonstrated in a real world organization (chapter 6) and there were no delays nor additional costs.

7.3 Questionnaire

A form of evaluating IS artifacts is to perform a quantitative analysis of our proposal, which results in a measured or perceived numeric value [39]. To achieve this, we used a questionnaire.

The questionnaire's goal was to validate the correctness and utility of our reference model. It was shared with CMMI professionals and practitioners and 19 responses were collected. All subjects had a similar knowledge of CMMI.

7.3.1 Questionnaire structure

The questionnaire was composed of three sections and a total of ten questions. The full questionnaire can be found in Appendix D.

The first section assessed the subjects' CMMI knowledge.

The second section introduced the context of our proposal with a brief explanation of our reference model. The intent of this section was to evaluate our reference model in terms of efficacy, utility, and ease of use. Therefore, subjects were asked to agree with the presented sentences, using a rating scale with agreement levels from 1-*Totally Disagree* to 5-*Totally Agree*.

On the last section, subjects were asked to compare a textual description with a graphical model. A snippet of a textual description from the CMMI v2.0 Full Model [7] and the correspondent graphical representation in ArchiMate from our reference model, were presented in the questionnaire.

7.3.2 Questionnaire data analysis

When asked about the sentence "This is a good way to represent the framework", the majority of the subjects *agreed*. The results are shown in Fig. 7.2.

Regarding the sentence "A model with graphical elements can be useful for someone who is learning or using CMMI", the majority of the subjects *totally agreed*. The results are shown in Fig. 7.3.

As for the sentence "The reference model presented could facilitate the use of CMMI", the majority of the subjects *agreed*. The results are shown in Fig. 7.4.

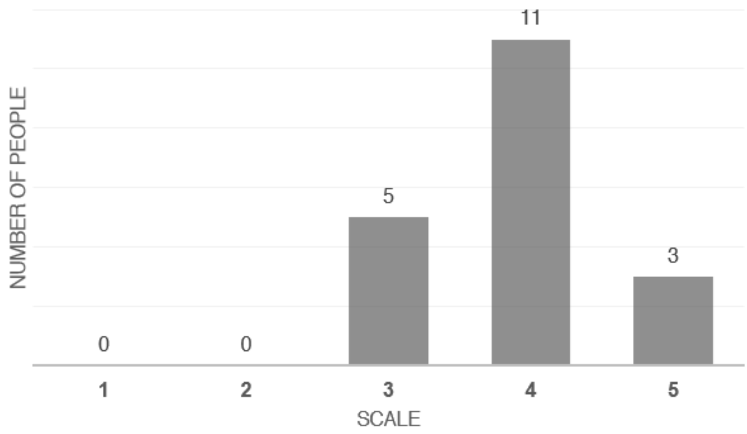


Figure 7.2: Responses to the sentence ”This is a good way to represent the framework”

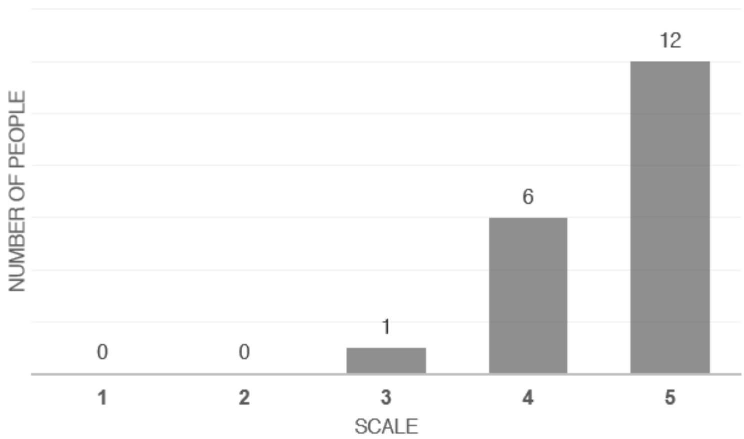


Figure 7.3: Responses to the sentence ”A model with graphical elements can be useful for someone who is learning or using CMMI”

To summarize and analyze the reliability of the collected data, Fig. 7.5 presents the mean results of the questionnaire regarding the three previous questions, with a 95% Confidence Interval. Q1 refers to the sentence ”This is a good way to represent the framework”, Q2 refers to the sentence ”A model with graphical elements can be useful for someone who is learning or using CMMI”, and Q3 refers to the sentence ”The reference model presented could facilitate the use of CMMI”.

Regarding the last section of the questionnaire, when asked to compare a textual description with a graphical model and answer the question ”Which representation do you find easier to read?”, 63% answered ”Graphical model in ArchiMate”.

Some of the reasons given to justify the choice of ”Graphical Model in ArchiMate” were:

- ”More friendly.”;
- ”Any model represented graphically is easier to understand than in text.”;
- ”I am a visual person meaning that I can capture easily a perception instead of reading and re-

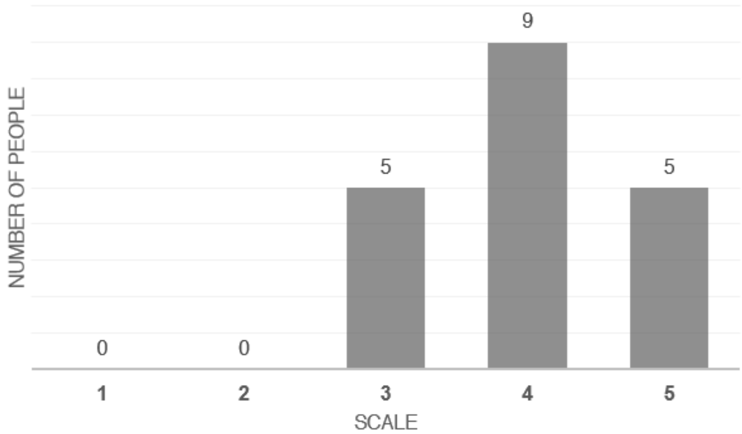


Figure 7.4: Responses to the sentence ”The reference model presented could facilitate the use of CMMI”

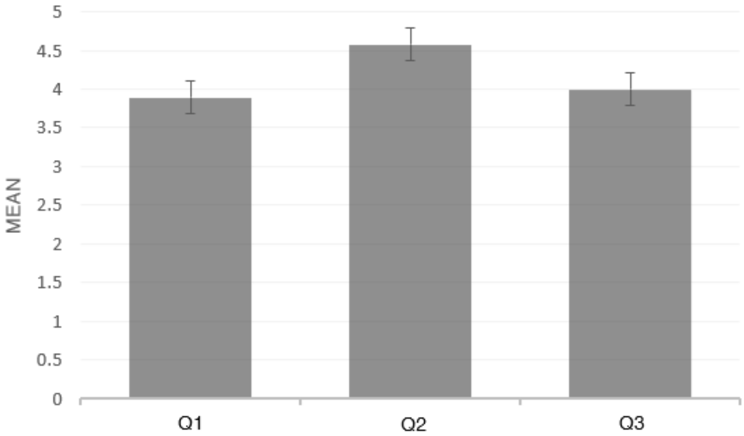


Figure 7.5: Mean results of the questionnaire with a 95% Confidence Interval

membering. Also, graphics are easier to memorize.”;

- ”It is more appealing and intuitive, which uncomplicates the process of understanding the model.”.

On the other hand, some of the reasons given to justify the choice of ”Textual description from the CMMI Model v2.0 Manual” were:

- ”Older people, like me, will gravitate towards the textual display and younger people would probably prefer the graphical approach.”;
- ”Text representation is familiar to me and is easier to read. Although, graphical model proposed gives interesting additional information.”.

By analyzing the questionnaire results, in general, the collected feedback was positive. This leads us to believe that our proposal is valid and tackles the complexity of CMMI. The subjects believed that this is a good way to represent this framework and that graphical models, such as ours, are useful and facilitate the use of CMMI.

With the feedback collected, we made some changes in our reference model to improve it. Those changes are mentioned in section 5.2.2.

7.4 Interview

Demonstrations are considered an early evaluation activity [39]. Following the demonstration done in a real-world organization, we interviewed relevant stakeholders in this organization to validate the value and correctness of our proposal, as well as verify if the demonstration of our proposal helped to solve the identified problem.

We interviewed two people from the company, the Quality Assurance Director and the Test Team Leader.

7.4.1 Interview structure

The interview was conducted by one author with both interviewees at the same time. It was a semi-structured interview, divided in three parts, with the duration of one hour. Being a semi-structured interview, it allowed for a discussion with the interviewees.

The interview started by showing our proposed metamodel of CMMI v2.0 (section 5.2.1) and the metamodel of CMMI-DEV v1.3 from Valverde et al. [12]. The intent was to see what changed in terms of the concepts from CMMI-DEV v1.3 to CMMI v2.0.

Following the metamodel, we presented our proposed CMMI v2.0 reference model (section 5.2.2), explaining the different views. Afterwards, we asked questions regarding the reference model.

The second part of the interview concerned the mappings. We showed the mappings between the company's procedures and CMMI-DEV v1.3 (section 6.1), the mappings between CMMI-DEV v1.3 and CMMI v2.0 Practices (section 5.2.3), and the mappings between the company's procedures and CMMI v2.0 (section 6.2). We then asked questions regarding these mappings.

The final part consistent on presenting the models of the company's TO-BE EA (section 6.3). Using ArchiMate's migration layer, we showed what processes and artifacts the company needs to add to be compliant with CMMI v2.0. Afterwards, we asked questions regarding these models.

7.4.2 Interview results

Table 7.1 shows the questions that were asked regarding our reference model of CMMI v2.0 and the respective answers from the interviewees.

Table 7.2 shows the questions that were asked regarding the mappings and the respective answers from the interviewees.

Table 7.1: Interviewees' answers regarding the reference model

Questions	Answers
Can this CMMI v2.0 reference model facilitate the use of CMMI?	"Being able to navigate throughout the whole model is awesome, it is very interesting to navigate the whole structure." "It really facilitates the use of CMMI and allows us to have a macro view of the structure."
Can a model with graphical elements be useful for someone who is learning or using CMMI? Would this reference model be useful in training sessions?	"This is especially useful for training sessions, so that people can have a better understanding of the practices." "Yes, very useful. Before, we were only able to show snippets of the model and mainly in text format, now we can see all the elements."

Table 7.2: Interviewees' answers regarding the mappings

Questions	Answers
Can these mappings, between your company's AS-IS EA and CMMI, facilitate the migration? Does it allow you to check what changes need to be done?	"Although now we are seeing this in a theoretical way, the mapping that was done already helps to give a more practical vision of what is missing. I liked the way it was presented with the colors." "Even for someone that does not have a lot of knowledge about CMMI, it is easy to understand the mapping and see where we might need to intervene. We get a clear vision of what is missing because of the dashed circles."

Finally, table 7.3 shows the questions that were asked regarding the TO-BE models and the respective answers from the interviewees.

The feedback given in the interview was very positive. For them, our CMMI v2.0 reference model is a good way to represent the framework and is especially useful in training sessions. The mappings with the colors and the dashed circles allowed them to clearly see what changes need to be done. Also, being able to see the whole model and the whole path, from beginning to end, makes it easier to follow and to understand the migration. Thereby, we can conclude that our proposal was useful and will help this company migrate from CMMI-DEV v1.3 to CMMI v2.0.

Table 7.3: Interviewees' answers regarding the TO-BE models

Questions	Answers
Does this model of your company's AS-IS and TO-BE EA allow you to clearly see what changes occurred with the migration to CMMI v2.0?	”The added value of this analysis is for us to be able to easily see the changes we need to make in the company's procedures. Because we see the workflow, we can see what we need to change.” ”This model of the AS-IS and the TO-BE is simple and understandable, therefore we can easily understand what we have now versus what we have to do. We have the vision from start to finish.”
Is this useful for the organization and for the people?	”There is always going to be resistance to change but this is an easy way to explain to people what we have now versus what we wish our procedures to be. For me, this uncomplicated CMMI and the comparison of one version with the other, which was something that scared me.” ”Here we have the whole path in one place and people will not get lost. Also, if we needed to show this to the administration, they would understand the migration and they would like this.”

8

Conclusion

Contents

This thesis was done by following DSRM which comprises 6 phases of development. First, the research problem was identified as being the difficulty to migrate from CMMI-DEV v1.3 to CMMI v2.0. Then, we defined the main objective of our proposal, which is to facilitate the migration for companies that are already CMMI-DEV v1.3 accredited and want to migrate to CMMI v2.0 by providing a reference model in ArchiMate to help this transition.

To solve the problem identified, we did a concept mapping between CMMI and ArchiMate and developed a metamodel. We then instantiated the metamodel and created our CMMI v2.0 reference model. Additionally, we did a visual mapping between CMMI-DEV v1.3 and CMMI v2.0 Practices.

The utility of our model was demonstrated in a real-world organization that was already CMMI-DEV v1.3 accredited and needed to migrate to CMMI v2.0. Our demonstration was done using the ArchiMate modelling language and the BiZZdesign Enterprise Studio modelling tool, and consisted on mapping company's procedures with CMMI-DEV v1.3, mapping company's procedures with CMMI v2.0, and modelling the transition.

We evaluated our proposal using a questionnaire to experts and practitioners and an interview with stakeholders of the organization in which we did our demonstration, as well as other well-known techniques to evaluate DS artifacts.

With the collected feedback, we proved our proposal is easy to understand, easy to use, and useful for the organization, facilitating the migration, as well as making CMMI more understandable.

The following sections describe our thesis' communication, contributions, limitations, as well as possible future work.

8.1 Communication

This section corresponds to the DSRM's communication phase and consists in communicating to the scientific community the utility and effectiveness of our proposed artifacts.

This document itself is part of this communication. It will be submitted and presented to a qualified jury, leading to a discussion and an evaluation, and later be made public.

Furthermore, we submitted a complete research paper to the ECIS 2020 conference for the track Enterprise Modelling and are waiting to know if it was accepted. We will receive an answer in the beginning of 2020.

8.2 Contributions

The main contributions of this thesis are: (1) a concept mapping between CMMI v2.0 and ArchiMate; (2) a CMMI v2.0 reference model in ArchiMate which provides a visual and structured representation of

CMMI v2.0; (3) A visual mapping between CMMI-DEV v1.3 and CMMI v2.0.

With those artifacts we hope to have addressed the research problem, aiding the migration from CMMI-DEV v1.3 to CMMI v2.0, as well providing organizations a better understanding of CMMI v2.0.

8.3 Limitations

As for limitations, since CMMI v2.0 is so recent and all the information is provided by the CMMI Institute, it was difficult to find different sources of information because there are not many researches about it yet.

Regarding the demonstration, the company we chose to do our demonstration with, was going through several changes in regard to their procedures and tools, that resulted in them delaying their migration to CMMI v2.0. These changes also caused a delay in our work since they were going through a transition period and we had to decide which state of the company we were going to model.

Furthermore, the models of the company's EA and the mappings with CMMI were done based on the documentation provided by the company, which may not represent the actual EA.

Another limitation we found regards this thesis being a continuation of a thesis about a CMMI-DEV v1.3 reference model [12]. CMMI-DEV v1.3 was more complex, had more concepts, and more layers. With CMMI v2.0, the CMMI Institute tackled some of those problems by simplifying the structure of CMMI, having less concepts and only three layers. Therefore, although the graphical model is still useful to aid the use and understanding of CMMI v2.0, it is not as useful as it was for CMMI-DEV v1.3. Additionally, CMMI-DEV v1.3 concepts were more aligned with ArchiMate.

8.4 Future work

Regarding future work, it would be interesting to demonstrate and evaluate our proposal in more organizations, as well as do the demonstration for other CMMI practice areas.

Additionally, it would be useful to automate the mapping between CMMI Practices and the company's EA. For this to be possible, an algorithm needs to be created. The information could be extracted from a matrix containing which company procedures answer the CMMI Practices.

Lastly, to show the advantages of migrating to CMMI v2.0 it would be interesting to find evidences of improvement in a real project. Resistance to change is an important factor to consider and if people see the advantages the migration can bring, they are possibly going to be more open to this change. Therefore, by showing the impact of CMMI in a project compliant with CMMI-DEV v1.3 Practices versus the impact if it was compliant with CMMI v2.0 Practices, we can see the differences and find evidences of improvement.

Bibliography

[1] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[2] The Open Group, "TOGAF® Standard, Version 9.2," 2018. [Online]. Available: http://pubs.opengroup.org/architecture/togaf9-doc/arch/

[3] ——, "Relationship to Other Standards," 2017. [Online]. Available: https://pubs.opengroup.org/architecture/archimate3-doc/apdxd.html#_Toc489946186

[4] ——, "ArchiMate® 3.0.1 Specification," 2017. [Online]. Available: http://pubs.opengroup.org/architecture/archimate3-doc/

[5] Y. Wand and R. Weber, "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars," *Information Systems Journal*, vol. 3, no. 4, pp. 217–237, 1993.

[6] G. O'Regan, *Concise Guide to Software Engineering: From Fundamentals to Application Methods*. Springer, 2017.

[7] CMMI Product Team, *CMMI Model V2.0*. CMMI Institute, 2018.

[8] D. Goldenson and D. Gibson, "Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results," *Carnegie Mellon University*, October 2013.

[9] M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, "An exploratory study of why organizations do not adopt CMMI," *Journal of Systems and Software*, vol. 80, no. 6, pp. 883–895, June 2007.

[10] N. Khurshid, P. Bannerman, and M. Staples, "Overcoming the First Hurdle: Why Organizations Do Not Adopt CMMI," *Springer*, 2009.

[11] D. B. Huang and W. Zhang, "CMMI in Medium & Small Enterprises: Problems and Solutions," *2nd IEEE International Conference on Information Management and Engineering*, 2010.

[12] L. Valverde, M. M. da Silva, and M. R. Gonçalves, "CMMI-DEV v1.3 Reference Model in ArchiMate," Instituto Superior Técnico, Portugal, 2018.

[13] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[14] M. Niazi, "A comparative study of software process improvement implementation success factors," *Journal of Software: Evolution and Process*, vol. 27, no. 9, 2015.

[15] Jeff Dalton, "Adopting the CMMI - How hard is it?" 2012. [Online]. Available: http://askthecmmiappraiser.blogspot.com/2012/06/adopting-cmmi-how-hard-is-it.html

[16] John Powell, "Survey of the SEI / Carnegie Mellon CMMI framework," 2012. [Online]. Available: http://www.umsl.edu/~sauterv/analysis/6840papers_f12/Powell/index.html

[17] CMMI Product Team, *CMMI for Development, Version 1.3*. Software Engineering Institute, November 2010.

[18] CMMI Institute, "When Will CMMI V1.3 Sunset And No Longer Be Supported?" 2018. [Online]. Available: https://cmmiinstitute.zendesk.com/hc/en-us/articles/360000701547-When-will-CMMI-V1-3-sunset-and-no-longer-be-supported-

[19] M. Lankhorst, *Enterprise architecture at work: Modelling, Communication, and Analysis*. Springer, 2017.

[20] CMMI Institute, "About CMMI® Institute," 2018. [Online]. Available: https://cmmiinstitute.com/company

[21] ——, "Appraisals," 2018. [Online]. Available: https://cmmiinstitute.com/learning/appraisals

[22] ——, "How Is The Terminology Changing In CMMI V2.0?" 2018. [Online]. Available: https://cmmiinstitute.zendesk.com/hc/en-us/articles/360000711887-How-is-the-terminology-changing-in-CMMI-V2-0

[23] T. Tamm, P. B. Seddon, G. Shanks, and P. Reynolds, "How Does Enterprise Architecture Add Value to Organisations?" *CAIS*, vol. 28, pp. 141–168, 2011.

[24] U. Franke, M. Ekstedt, R. Lagerström, J. Saat, and R. Winter, "Trends in Enterprise Architecture Practice – A Survey," *International Workshop on Trends in Enterprise Architecture Research*, p. 16–29, 2010.

[25] J. Zachman, "The Zachman Framework Evolution," *EA Articles*, 2009.

[26] C. M. Pereira and P. Sousa, “A Method to Define an Enterprise Architecture using the Zachman Framework,” *Proceedings of the 2004 ACM Symposium on Applied Computing (SAC)*, 2004.

[27] Object Management Group, “MDA,” 2019. [Online]. Available: https://www.omg.org/mda/

[28] T. Kühne, “Matters of (meta-) modeling,” *Software & Systems Modeling*, vol. 5, no. 4, p. 369–385, 2006.

[29] M. Goeken and S. Alter, “Towards Conceptual Metamodeling of IT Governance Frameworks Approach - Use - Benefits,” *Proceedings of the 42nd Hawaii International Conference on System Sciences*, 2009.

[30] R. Schutte and T. Rotthowe, “The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models,” *International Conference on Conceptual Modeling*, pp. 240–254, 1998.

[31] G. H. Soydan and M. M. Kokar, “A partial formalization of the CMMI-DEV - A capability maturity model for development,” *Journal of Software Engineering and Applications*, 2012.

[32] D. Musat, V. Castaño, J. A. Calvo-Manzano, and J. Garbajosa, “MATURE: A Model Driven bAsed Tool to Automatically Generate a langUage That suppoRts CMMI Process Areas specification,” *European Conference on Software Process Improvement*, pp. 48–59, 2010.

[33] J. Teixeira, M. M. da Silva, and E. Carreira, “Modelling Risk Management using ArchiMate,” Instituto Superior Técnico, Portugal, 2017.

[34] M. Vicente, N. Gama, and M. M. da Silva, “Modeling ITIL Business Motivation Model in ArchiMate,” *Business Informatics (CBI), IEEE 15th Conference*, pp. 270–275, 2013.

[35] N. Silva, M. M. da Silva, B. Barafort, M. Vicente, and P. Sousa, “Using ArchiMate to Model a Process Assessment Framework,” *In Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pp. 1189–1194, April 2015.

[36] R. Almeida, P. L. Pinto, and M. M. da Silva, “Using ArchiMate to Assess COBIT 5 and ITIL Implementations,” *25th International Conference on Information Systems Development, Poland*, 2016.

[37] I. Percheiro, R. Almeida, P. L. Pinto, and M. M. da Silva, “Towards Conceptual Meta-Modeling of ITIL and COBIT 5,” *In European, Mediterranean, and Middle Eastern Conference on Information Systems*, pp. 478–491, 2017.

[38] D. L. Moody and G. Shanks, “Improving the Quality of Data Models: Empirical Validation of a Quality Management Framework,” *Information Systems*, vol. 28, no. 6, pp. 619–650, 2003.

[39] N. Prat, I. Comyn-Wattiau, and J. Akoka, “Artifact Evaluation in Information Systems Design Science Research - A Holistic View,” *PACIS 2014 Proceedings*, vol. 23, p. 1–16, 2014.

A

Appendix A: Full Reference Model

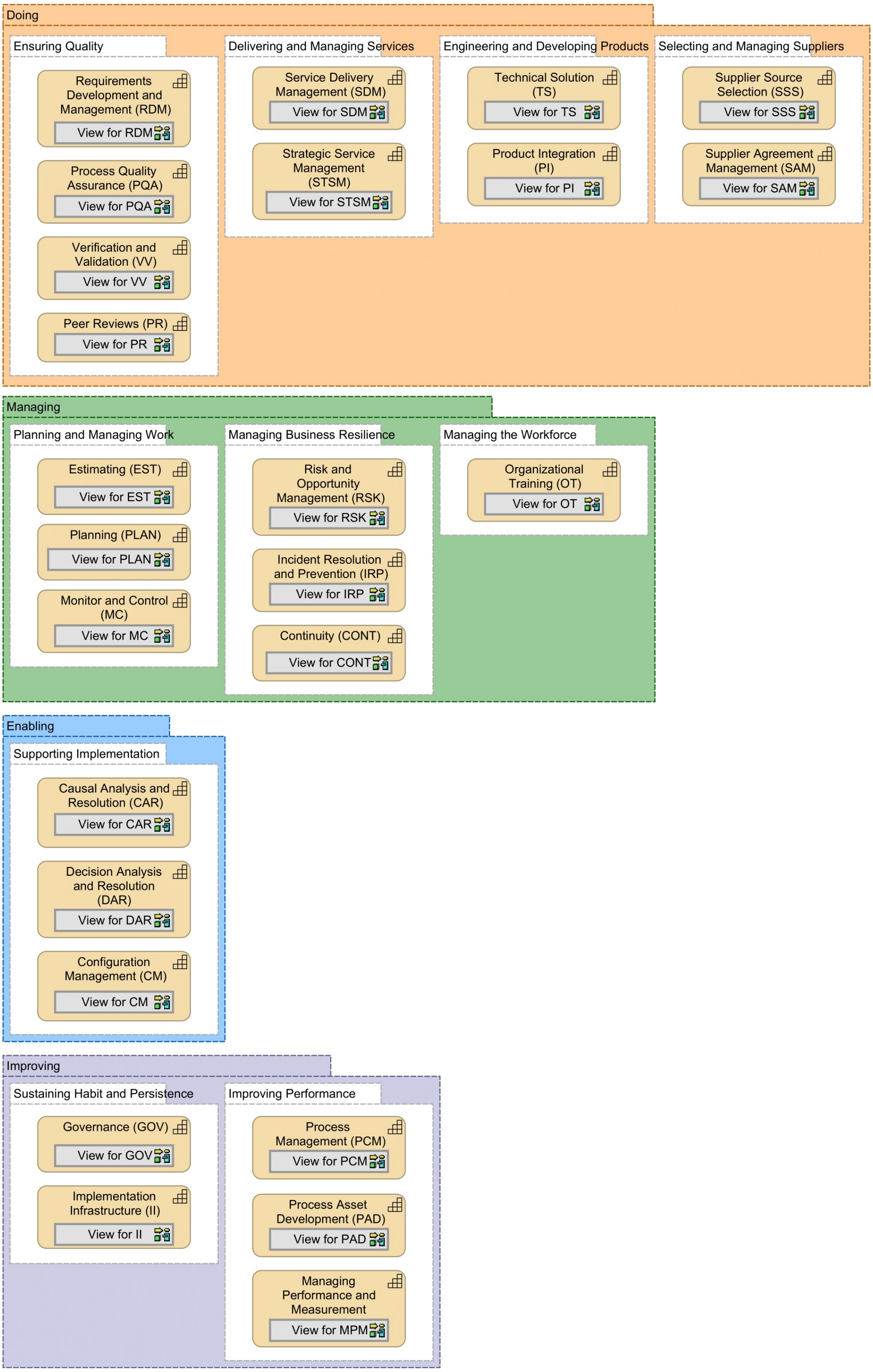


Figure A.1: Full first level of abstraction of the reference model

B

Appendix B: Mappings between company's procedures and CMMI-DEV v1.3

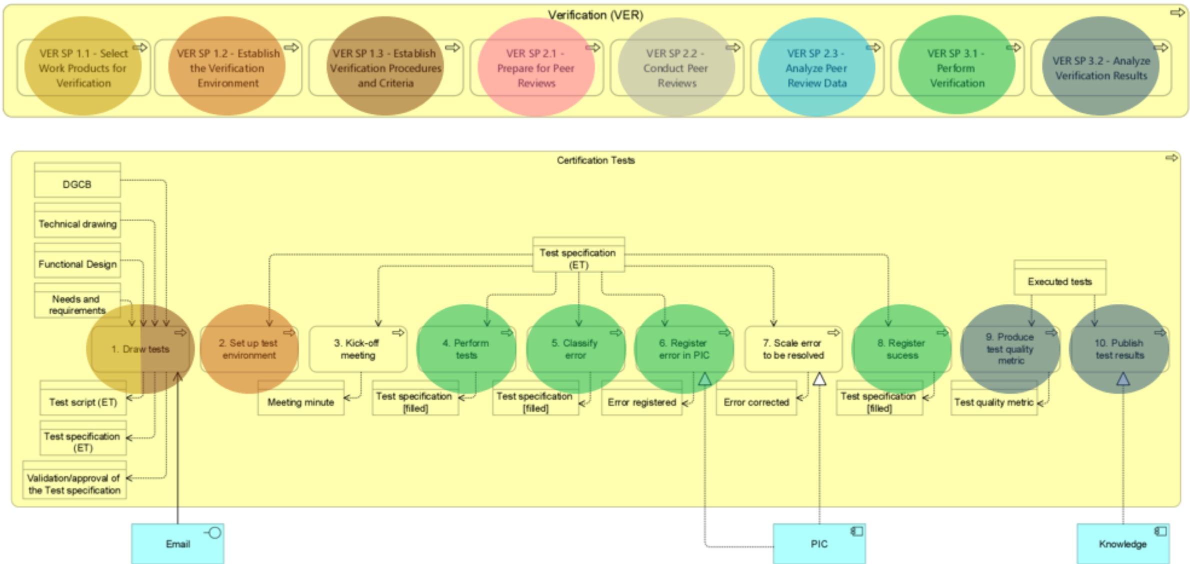


Figure B.1: Mapping between VER specific practices and Certification Tests procedure

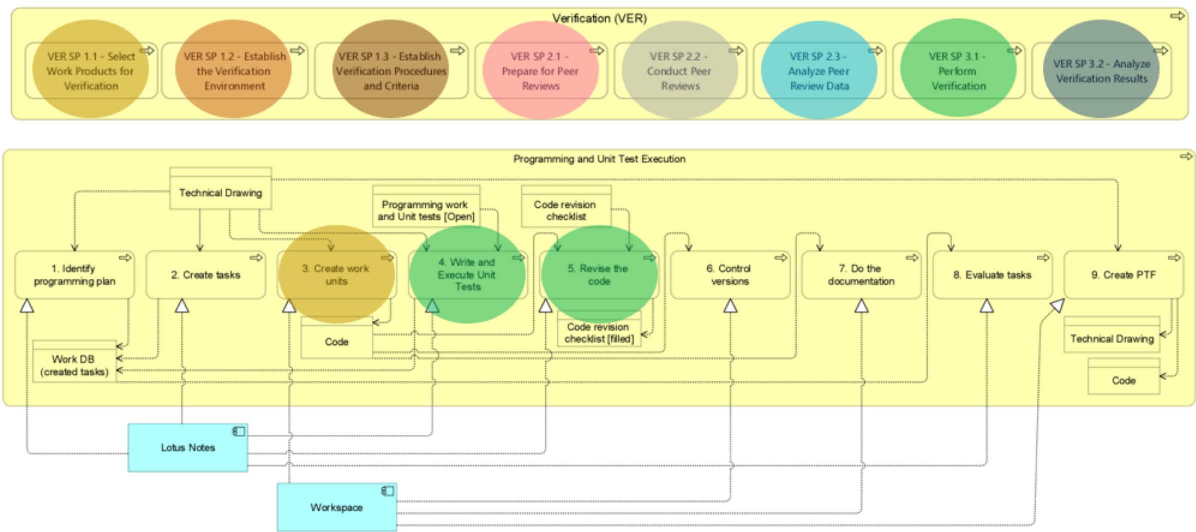


Figure B.2: Mapping between VER specific practices and Unit Tests procedure

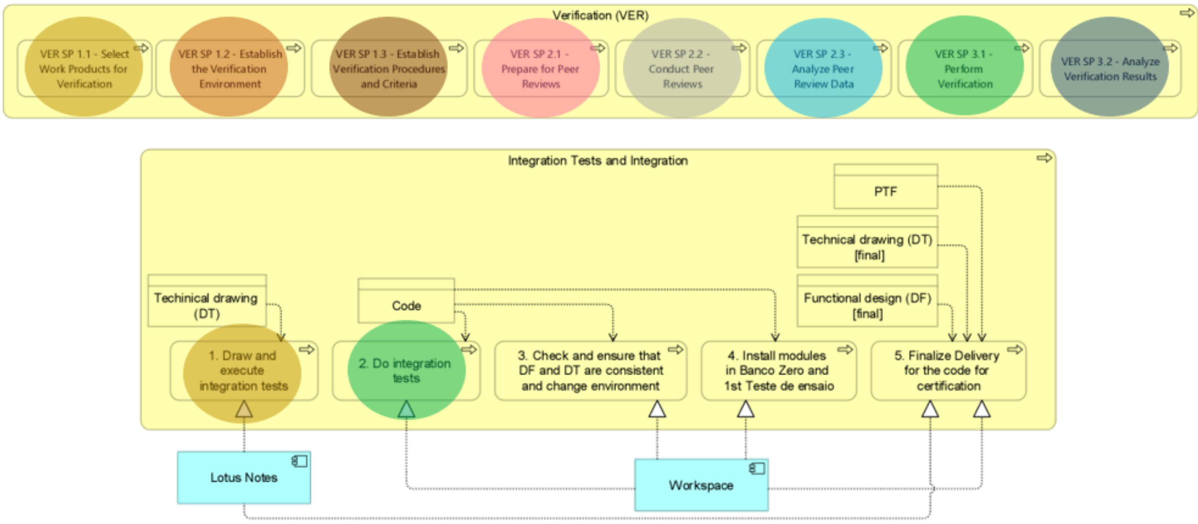


Figure B.3: Mapping between VER specific practices and Integration Tests procedure

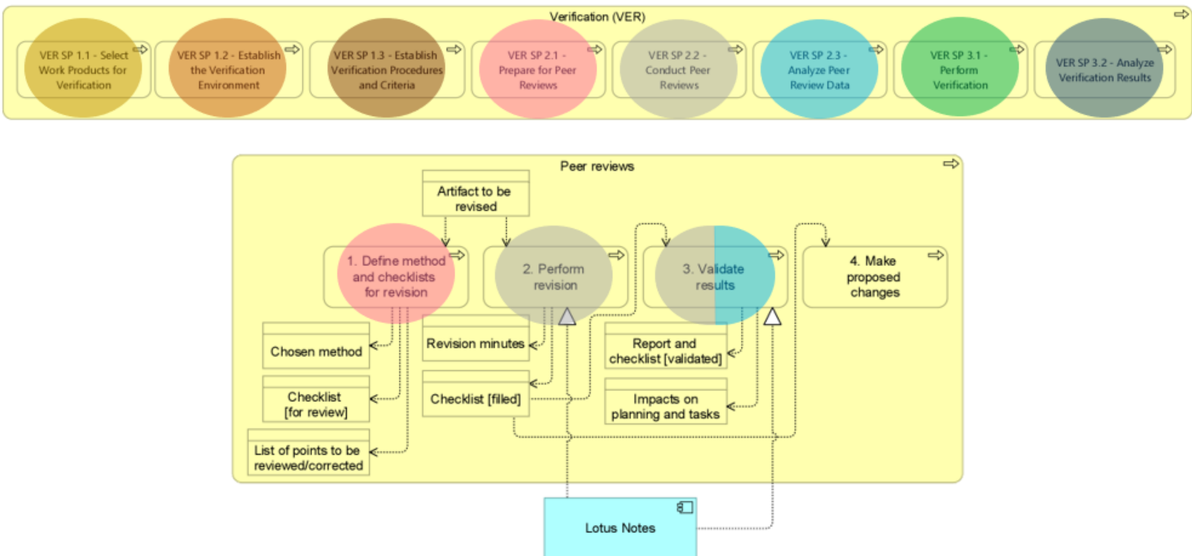


Figure B.4: Mapping between VER specific practices and Peer Reviews procedure

C

Appendix C: Mappings between company’s procedures and CMMI v2.0

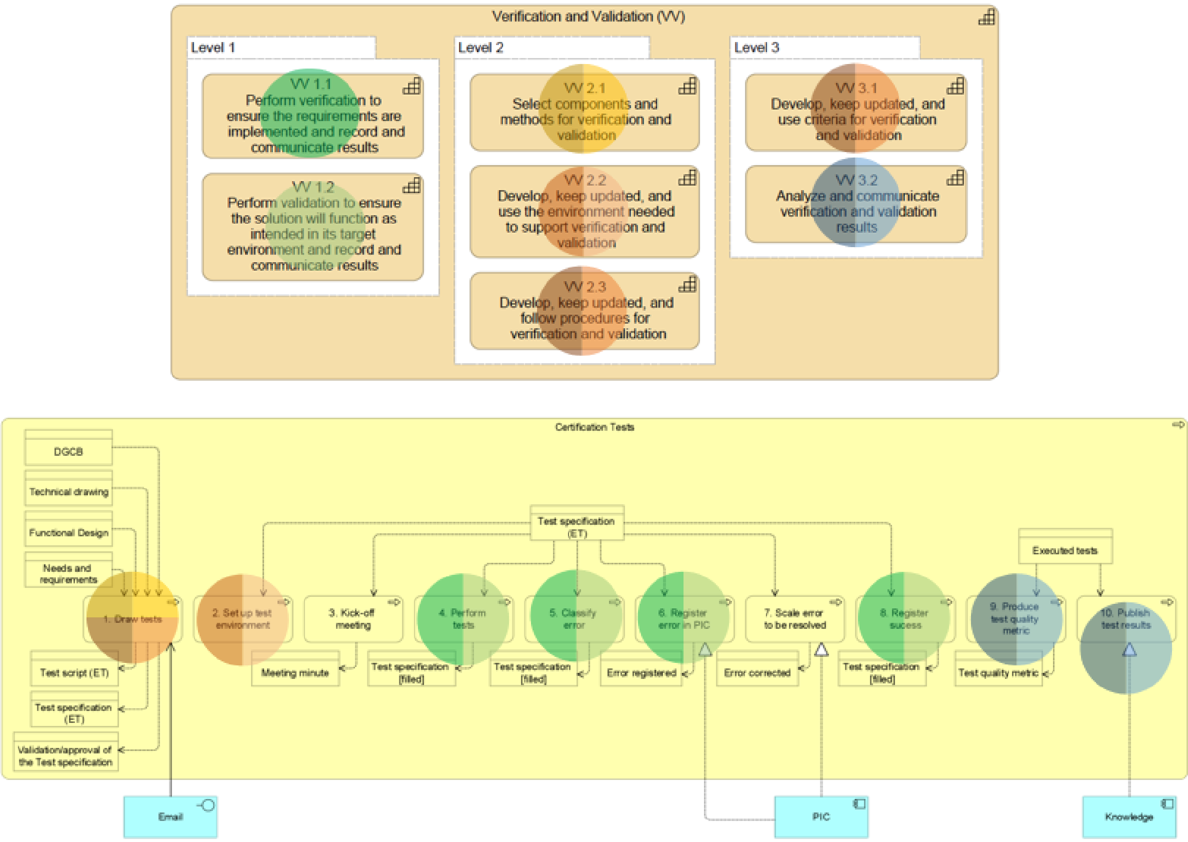


Figure C.1: Mapping between VV Practices and Certification Tests procedure

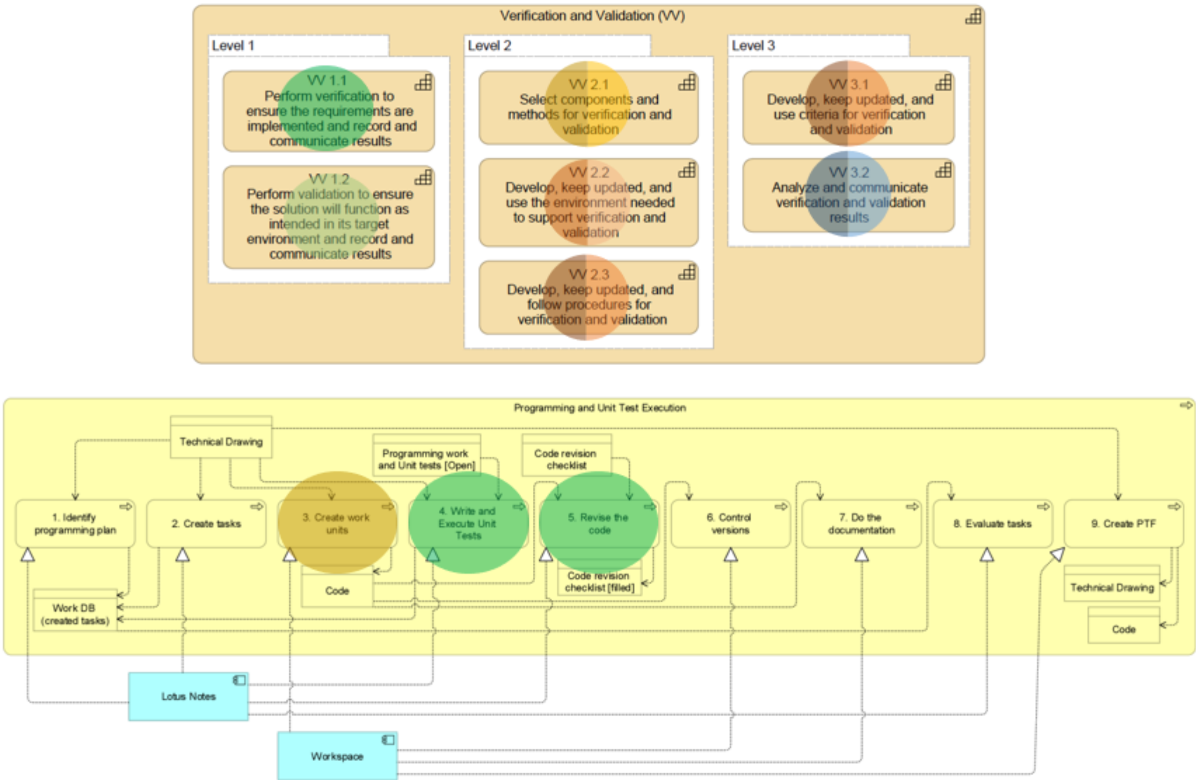


Figure C.2: Mapping between VV Practices and Unit Tests procedure

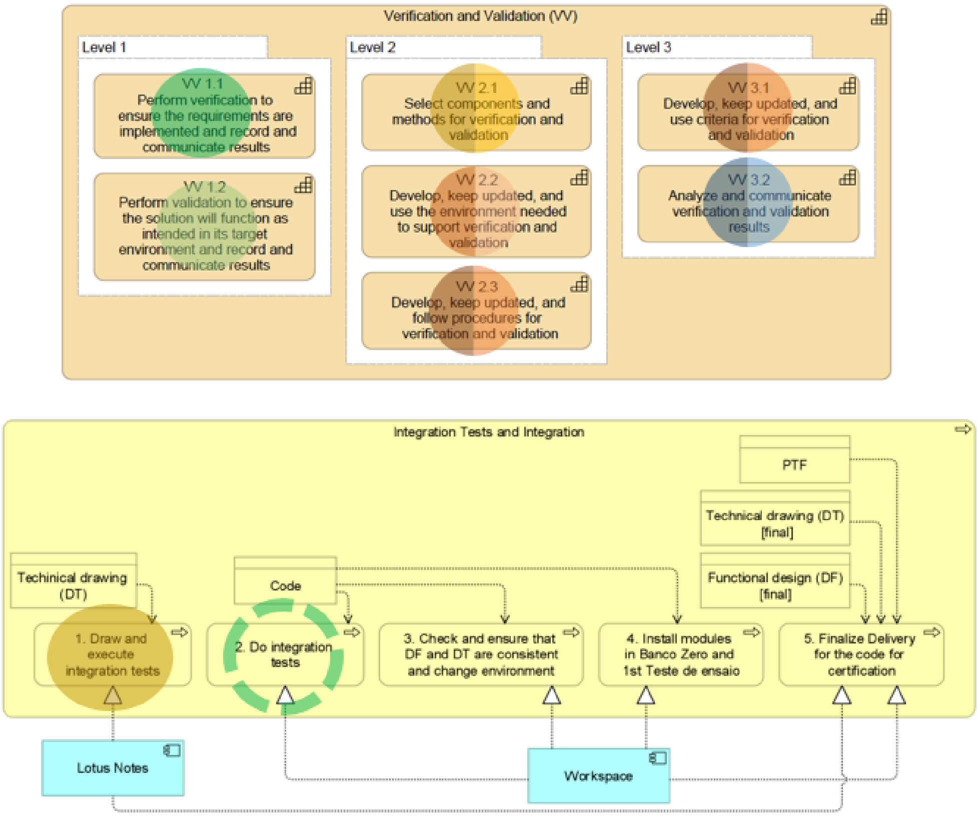


Figure C.3: Mapping between VV Practices and Integration Tests procedure

D

Appendix D: Questionnaire

CMMI V2.0 Reference Model Feedback

This questionnaire is intended for CMMI experts and practitioners.

The purpose of the questionnaire is to collect feedback on a CMMI V2.0 Reference Model in ArchiMate developed within the scope of a master thesis from Instituto Superior Técnico (Lisbon, Portugal).

This questionnaire will not take you longer than 5 minutes and is completely anonymous.

Your feedback will help improve our research and, consequently, facilitate the use of the CMMI framework. This data will be used solely for academic and scientific study purposes.

In case of any question, suggestion or problem related to this questionnaire, please contact mariana.s.silva@tecnico.ulisboa.pt.

Disclaimer: We have permission from the CMMI Institute to share this content.

Thank you for your time.

*Required

Knowledge assessment

1. **How familiar are you with CMMI and its terms?** *

   Mark only one oval.

	1	2	3	4	5	
Not familiar	⬭	⬭	⬭	⬭	⬭	Very familiar

2. **How familiar are you with CMMI Model V2.0 and its terms?** *

   Mark only one oval.

	1	2	3	4	5	
Not familiar	⬭	⬭	⬭	⬭	⬭	Very familiar

3. **In your opinion, how complicated is it to understand CMMI?** *

   Mark only one oval.

	1	2	3	4	5	
Not complicated	⬭	⬭	⬭	⬭	⬭	Very complicated

4. **How do you classify yourself?** *

Mark only one oval.

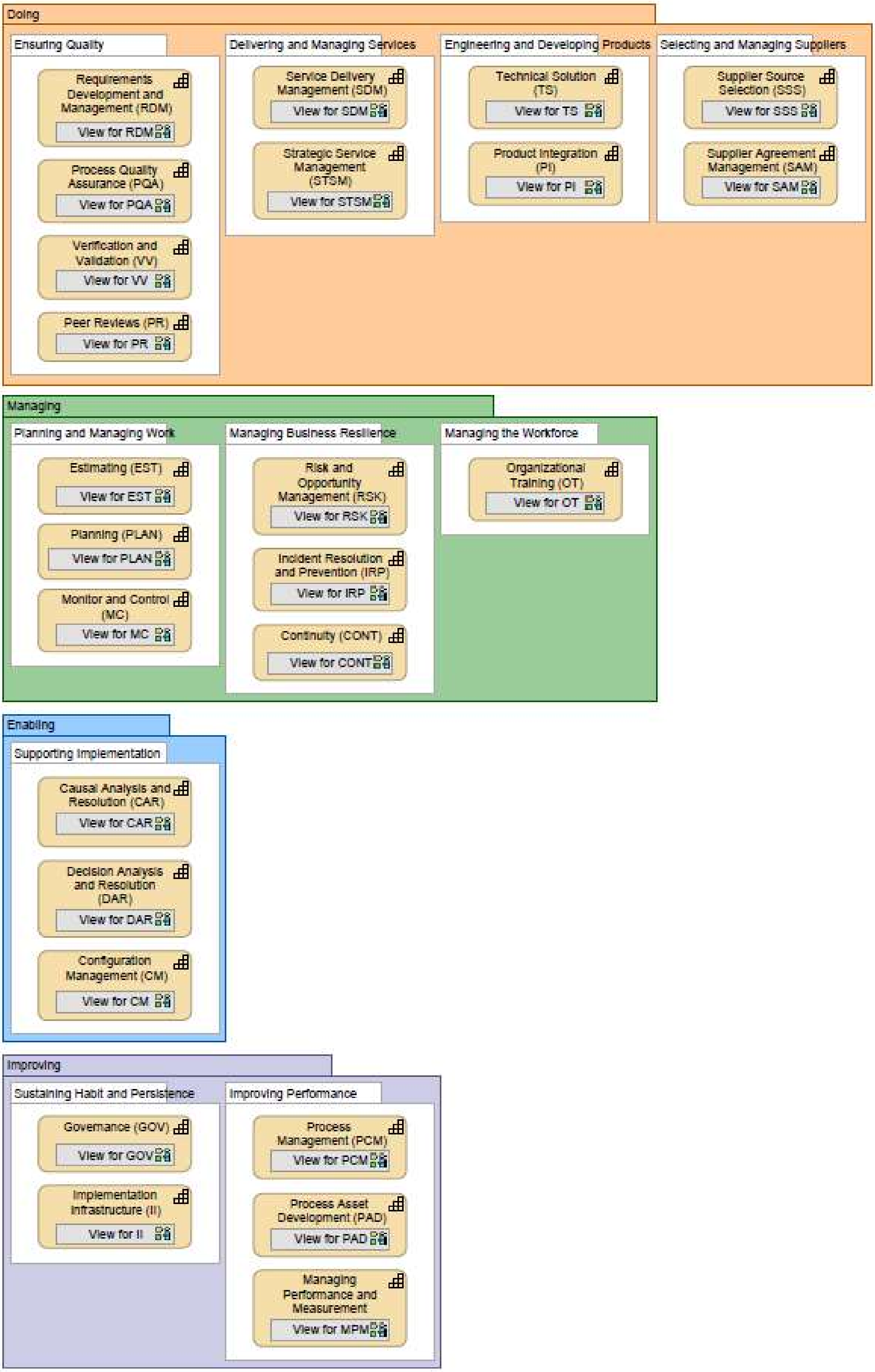
- Consultant
- CMMI professional
- Staff from an organization that is adopting or has adopted CMMI
- Professor
- Other: ______________________

CMMI V2.0 Reference Model

We propose a CMMI V2.0 reference model, designed in ArchiMate (a modelling language for enterprise architectures), using the modeling tool BiZZdesign Enterprise Studio.

This is an interactive model with graphical elements that shows CMMI V2.0 concepts and their relations.

The image bellow shows the model's first level of abstraction. It contains the Categories (Doing, Managing, Enabling and Improving) and these Categories contain the Capability Areas. Each Capability Area then contains a set of Practice Areas.

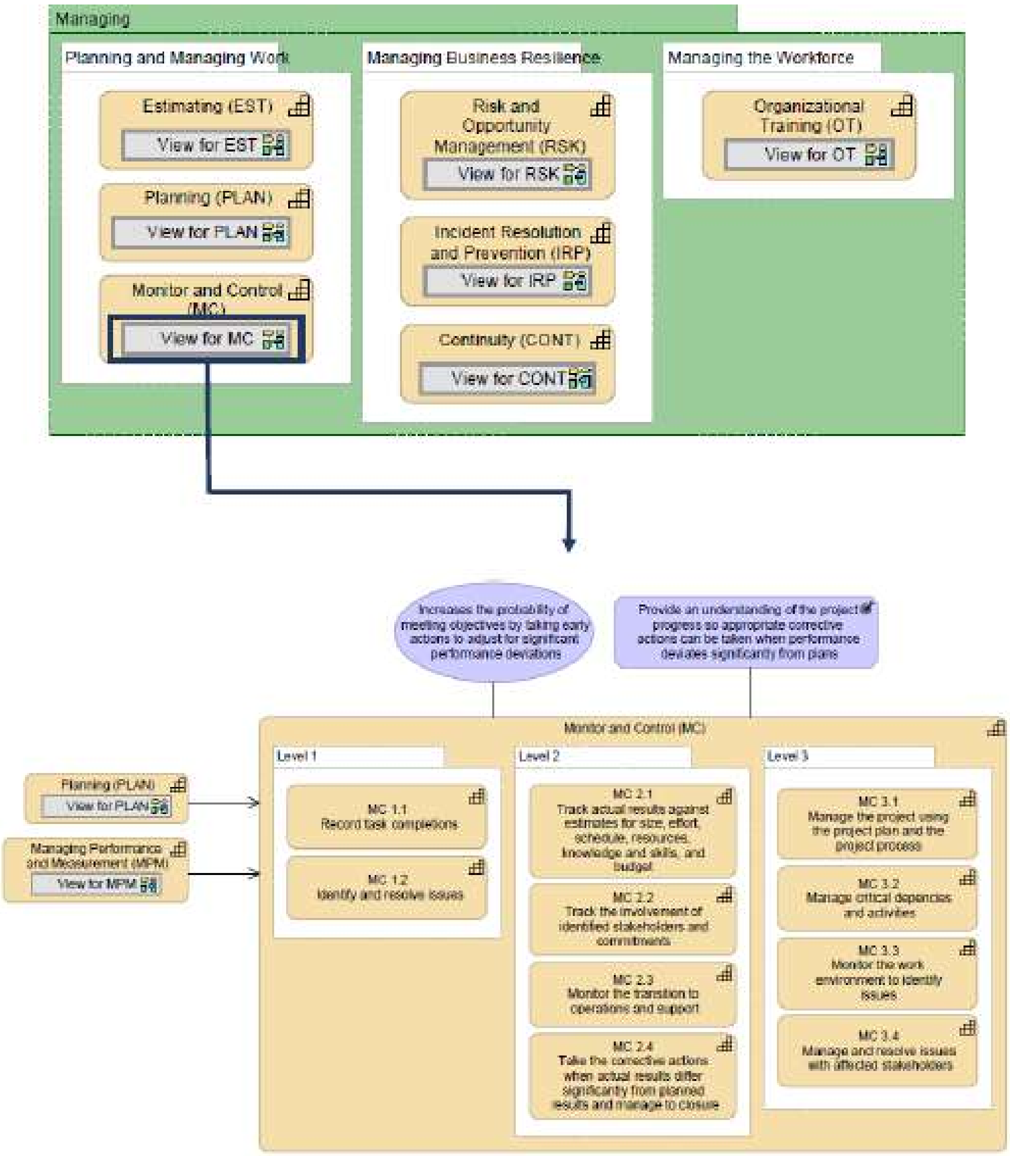


Doing
Ensuring Quality
Requirements Development and Management (RDM)
View for RDM
Process Quality Assurance (PQA)
View for PQA
Verification and Validation (VV)
View for VV
Peer Reviews (PR)
View for PR
Delivering and Managing Services
Service Delivery Management (SDM)
View for SDM
Strategic Service Management (STSM)
View for STSM
Engineering and Developing Products
Technical Solution (TS)
View for TS
Product Integration (PI)
View for PI
Selecting and Managing Suppliers
Supplier Source Selection (SSS)
View for SSS
Supplier Agreement Management (SAM)
View for SAM
Managing
Planning and Managing Work
Estimating (EST)
View for EST
Planning (PLAN)
View for PLAN
Monitor and Control (MC)
View for MC
Managing Business Resilience
Risk and Opportunity Management (RSK)
View for RSK
Incident Resolution and Prevention (IRP)
View for IRP
Continuity (CONT)
View for CONT
Managing the Workforce
Organizational Training (OT)
View for OT
Enabling
Supporting Implementation
Causal Analysis and Resolution (CAR)
View for CAR
Decision Analysis and Resolution (DAR)
View for DAR
Configuration Management (CM)
View for CM
Improving
Sustaining Habit and Persistence
Governance (GOV)
View for GOV
Implementation Infrastructure (II)
View for II
Improving Performance
Process Management (PCM)
View for PCM
Process Asset Development (PAD)
View for PAD
Managing Performance and Measurement
View for MPM

In this reference model, inside each Practice Area we have the ArchiMate concept of view. Views help us navigate to other parts of the reference model and see an element in more detail.

For instance, if we click on “View for MC”, we can see the Practice Area “Monitor and Control (MC)” in more detail, as shown in the image bellow.

Inside the Practice Area “Monitor and Control (MC)” we have the Practices that compose the Practice Area grouped in levels (which, in this case, only go until Level 3), the value and intent of the Practice Area, and the Related Practice Areas.



Answer the following questions regarding the images above

5. **This is a good way to represent the framework.** *

Mark only one oval.

	1	2	3	4	5	
Totally disagree	◯	◯	◯	◯	◯	Totally agree

6. **A model with graphical elements can be useful for someone who is learning or using CMMI.** *
Mark only one oval.

	1	2	3	4	5	
Totally disagree	⬭	⬭	⬭	⬭	⬭	Totally agree

7. **The CMMI V2.0 reference model presented above could facilitate the use of CMMI.** *
Mark only one oval.

	1	2	3	4	5	
Totally disagree	⬭	⬭	⬭	⬭	⬭	Totally agree

Please analyze the next two images

Both images show the intent, value, practice groups, and practices of the practice area Monitor and Control (MC).

Textual description from the CMMI Model V2.0 Manual

Intent

Provide an understanding of the project progress so appropriate corrective actions can be taken when performance deviates significantly from plans.

Value

Increases the probability of meeting objectives by taking early actions to adjust for significant performance deviations.

Additional Required PA Information

This section left blank for future content.

Explanatory PA Information

Practice Summary

Level 1

MC 1.1	Record task completions.
MC 1.2	Identify and resolve issues.

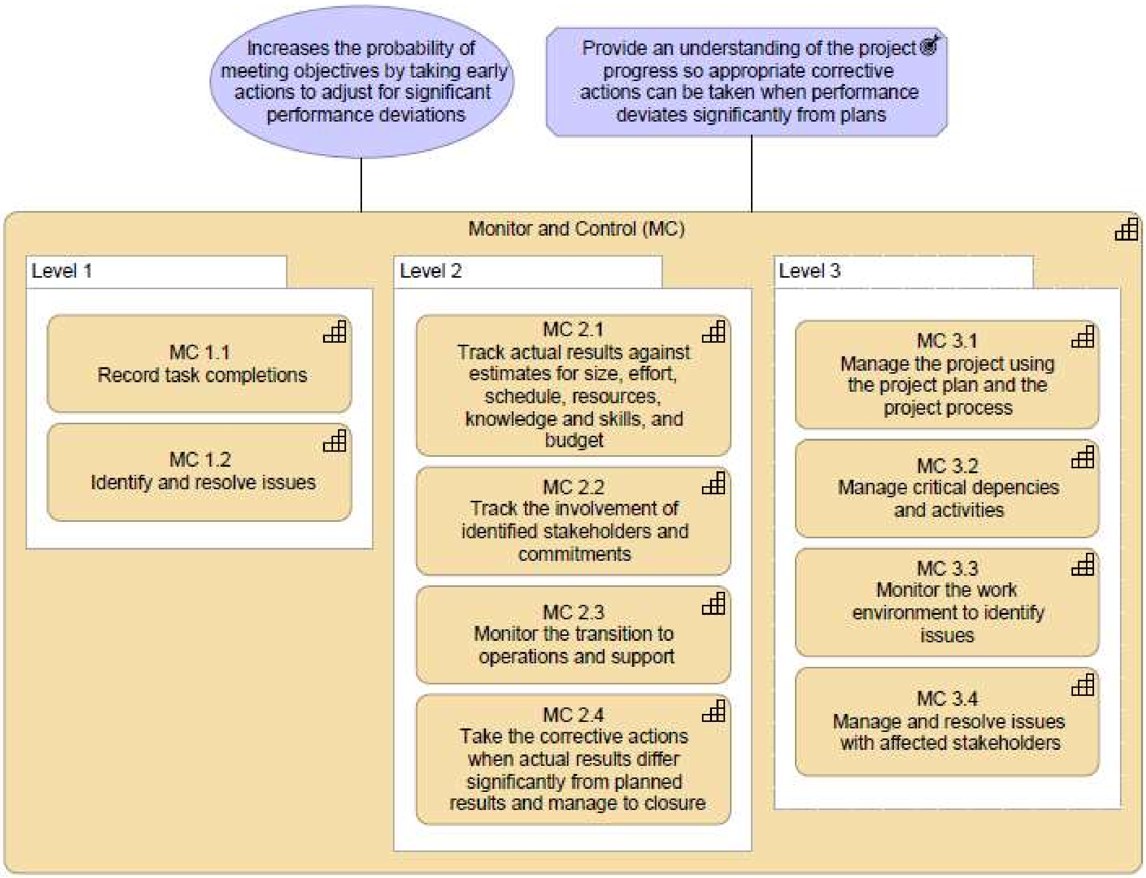
Level 2

MC 2.1	Track actual results against estimates for size, effort, schedule, resources, knowledge and skills, and budget.
MC 2.2	Track the involvement of identified stakeholders and commitments.
MC 2.3	Monitor the transition to operations and support.
MC 2.4	Take corrective actions when actual results differ significantly from planned results and manage to closure.

Level 3

MC 3.1	Manage the project using the project plan and the project process.
MC 3.2	Manage critical dependencies and activities.
MC 3.3	Monitor the work environment to identify issues.
MC 3.4	Manage and resolve issues with affected stakeholders.

Graphical model in ArchiMate



Answer the following questions regarding the images above

8. **Which representation do you find easier to read?** *

Mark only one oval.

- ◯ Textual description from the CMMI Model V2.0 Manual
- ◯ Graphical model in ArchiMate

9. **Can you justify your choice?**

Thank you for answering this questionnaire. Your feedback is very important.

10. **Feel free to leave comments, suggestions, ...**