CMMI-DEV v1.3 Reference Model in ArchiMate

Luís Miguel Gonçalves Valverde

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. António Manuel Ferreira Rito da Silva
Supervisor: Prof. Miguel Leitão Bignolas Mira da Silva
Members of the Committee: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

October 2018
Acknowledgments

In the first place, I would like to thank my parents, João and Ana for all the support these years, all their efforts make it possible for me to have an education and a better future, without their support, it would be impossible to be where I am now.

I would like to express my gratitude to my supervisor, Prof. Miguel Mira da Silva, who guide me during this research. His constant guidance, advice, and motivation were very important to make this research possible.

I would like to express my gratitude to my co-supervisor, Margarida Rolão Gonçalves, who help me a lot in understanding key points of this theme as well as being the bridge between me and the organization used in this research, it would be impossible without her help to validate this research the way I did it.

I would like to thank all my colleges from IST for all the share of knowledge and motivation, especially Gonçalo, Daniel and Bruno that where very important in the accomplishment of the Masters.

I would like to thank Andreia, who has always time to listen to my problems, even when she has her own problems, without her support many of the obstacles in my life would be more difficult to pass.

I would like to thank all the professionals from the organization used in this research for their time and availability and for let me go to their workshops regarding the CMMI project.

Last, but not least, I want to thank my brother, Pedro, for always being there when things are complicated.

Thank you all.
Abstract

To verify what are the existing concepts and their relationships in a framework, graphical reference models can be used instead of textual descriptions, giving a different and a clearer idea of how a framework works. Capability Maturity Model Integration (CMMI) is a well-known framework used by organizations to improve their processes. When adopting a framework like CMMI, organizations face many challenges, some of them related to the perceived complexity of the textual representation of CMMI by its users.

In order to reduce the high perceived complexity of CMMI by its users, a graphical reference model for CMMI using ArchiMate as the chosen Enterprise Architecture (EA) modeling language was proposed. The research is focused on the part of CMMI related to the development of both products and services, known as CMMI for Development (CMMI-DEV), in the version 1.3. With ArchiMate as the chosen EA modeling language, a reference model for CMMI-DEV v1.3 was developed following the Design Science Research Methodology (DSRM) with the purpose of reducing the high perceived complexity of this framework by representing its concepts and relationships with graphical concepts and relationships of ArchiMate.

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 the same version of CMMI-DEV used in this research. The demonstration in the research 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.

Finally, to validate both the proposed reference model and the demonstration, well-known techniques to evaluate Design Science (DS) artifacts (mappings, models), as well as interviews with the staff of the organization behind the project of adopting CMMI, were used. The research concludes with some findings and future work on this topic.

Keywords

Reference Models; Capability Maturity Model Integration; Enterprise Architecture; ArchiMate;
Resumo

Para verificar quais são os conceitos existentes e as suas relações numa framework, modelos de referência gráficos podem ser usados em vez de descrições textuais, dando uma ideia diferente e mais clara de como uma framework funciona. O CMMI é uma framework conhecida que é usada por organizações para melhorar os seus processos. Ao adotar uma framework como o CMMI, as organizações enfrentam muitos desafios, alguns deles relacionados com complexidade percebida da representação textual do CMMI pelos seus utilizadores.

A fim de reduzir a alta complexidade percebida do CMMI pelos seus utilizadores, foi proposto um modelo de referência gráfico para o CMMI usando o ArchiMate como a linguagem de modelação de Arquitetura Empresarial (AE) escolhida. O foco desta investigação está relacionado com a componente do CMMI focada no desenvolvimento de produtos e serviços, mais conhecido como CMMI-DEV na versão 1.3. Com o ArchiMate como a linguagem de modelação de AE escolhida, um modelo de referência para CMMI-DEV v1.3 foi desenvolvido com base na DSRM com o objetivo de reduzir a alta complexidade percebida desta framework representando os seus conceitos e relações com conceitos e relações gráficas do ArchiMate.

Para demonstrar a utilidade do modelo de referência CMMI-DEV v1.3 proposto, foi realizado um estudo de campo numa organização real que está a adotar a mesma versão do CMMI-DEV usada nesta investigação, com o intuito de melhorar os seus processos. Na demonstração, a AE da organização foi mapeada com o modelo de referência proposto com o objetivo de mostrar os potenciais benefícios de representar o CMMI com uma AE.

Finalmente, para validar tanto o modelo de referência proposto como a sua demonstração, técnicas bem conhecidas para avaliar artefatos de DS (mapeamentos, modelos) bem como entrevistas com a equipa da organização por de trás do projeto da adoção do CMMI foram usadas. A investigação conclui com algumas descobertas e trabalho futuro sobre este tópico.

Palavras Chave

Modelos de Referência; Capability Maturity Model Integration; Arquitetura Empresarial; ArchiMate;
## Contents

1 Introduction 1
   1.1 Research Problem ............................................. 2
   1.2 Research Methodology ...................................... 3
   1.3 Dissertation Outline ..................................... 5

2 Related Work 7
   2.1 Capability Maturity Model Integration ..................... 7
      2.1.1 CMMI for Development .............................. 8
      2.1.2 CMMI Staged Representation ....................... 9
   2.2 Enterprise Architecture ................................ 11
   2.3 ArchiMate .................................................. 13
   2.4 CMMI Ontologies .......................................... 14
   2.5 Information Systems Frameworks and ArchiMate .......... 15

3 Research Proposal 19
   3.1 Objectives of the Proposal ................................ 19
   3.2 Mapping CMMI-DEV v1.3 with ArchiMate ................... 19
   3.3 CMMI-DEV v1.3 Metamodel ................................ 21
   3.4 CMMI-DEV v1.3 Reference Model .......................... 22

4 Demonstration 27

5 Evaluation 35
   5.1 Artifact Evaluation in DSR ................................ 35
   5.2 Interviews .................................................. 38

6 Conclusion 41
   6.1 Limitations ................................................ 41
   6.2 Research Communication ............................... 42
   6.3 Future Work .............................................. 43

A Appendix - Parts of CMMI-DEV v1.3 Reference Model 49
List of Figures

1.1 DSRM mapped to our research. (Adapted from [Peffers et al., 2007]) .................. 4
2.1 TOGAF Architecture Development Method ...................................................... 12
2.2 Full ArchiMate Framework ............................................................................. 13
2.3 CMMI UML Ontology ................................................................................... 14
2.4 CMMI Ontology with the Representations ..................................................... 15
2.5 ISO 27001 and ISO TS 33052/33072 Metamodel ........................................... 16
3.1 CMMI-DEV v1.3 Metamodel in ArchiMate ..................................................... 21
3.2 Part of CMMI-DEV v1.3 Reference Model .................................................... 22
3.3 Basic relations of Project Management process areas .................................... 23
3.4 REQM process area model .......................................................................... 24
3.5 REQM specific practices and specific practice 1.2 models ............................ 25
3.6 REQM generic practices and generic practice 2.1 models ............................. 26
4.1 Mapping of VER process area with AS-IS state ......................................... 28
4.2 Mapping of VER process area with Definition phase of the Iterative Software Development process in the AS-IS state ....................................................... 29
4.3 Mapping of VER process area with TO-BE state ........................................ 30
4.4 Mapping of VER process area with Definition phase of the Iterative Software Development process in the TO-BE state ....................................................... 31
4.5 The two states of the EA with the proposed colors ........................................ 32
A.1 CMMI-DEV v1.3 Reference Model ............................................................. 49
A.2 Full REQM generic practices ................................................................. 50
List of Tables

2.1 CMMI-DEV v1.3 process areas ...................................................... 10
3.1 Mapping CMMI-DEV with ArchiMate ........................................... 20
5.1 Evaluation - interviewees opinions .............................................. 38
Acronyms

ADM Architecture Development Method
AE Arquitetura Empresarial
BS Behavioral Science
CMMI Capability Maturity Model Integration
CMMI-DEV CMMI for Development
COBIT Control Objectives for Information and Related Technologies
DS Design Science
DSR Design Science Research
DSRM Design Science Research Methodology
EA Enterprise Architecture
IS Information Systems
ISACA Information Systems Audit and Control Association
ISO International Organization for Standardization
IT Information Technologies
ITIL Information Technology Infrastructure Library
OWL Web Ontology Language
PMC Project Monitoring and Control
REQM Requirements Management
ROI Return on Investment
SEI Software Engineering Institute
SPI Software Process Improvement
TOGAF The Open Group Architecture Framework
VER Verification
1. Introduction

Nowadays, organizations are increasingly focusing on the quality and functionality of the software they develop, much of which involves redesigning their software processes to follow the best practices known in the industry so that the software they developed becomes trustworthy, reliable, of a high quality, fit its purpose and is consistently delivered on time to their customers [O’Regan, 2017].

A software process can be seen as the glue that connects people, tools and equipments, and procedures and methods in a consistent way through a set of interrelated activities that, together, interact to develop and maintain software and the associated work products [O’Regan, 2017, CMMI Product Team, 2010]. The Software Engineering Institute (SEI) that is an American research, development and training center involved in computer software and network security and other quality experts believe that there is a close relationship between the quality of the delivered software and the quality and maturity of the software processes [O’Regan, 2017]. To achieve mature processes with improved quality and effectiveness, the steps that organizations take are not so intuitive as they can do it without a certain type of guidance, therefore there was a need to create initiatives that lead to a focus on software processes and on ways to improve them [Fantina, 2005].

Software Process Improvement (SPI) initiatives define and measure best practices and processes for improving the existing processes in organizations, intending to help them develop higher quality software and products and achieve their business goals more efficiently, where the business goals can be: faster delivery of software and products to the market on time and budget; improved customer satisfaction; improved quality of software and products; and cost reduction of development. In other words, SPI helps organizations to work smarter, as they build software and products, better, faster and cheaper than the competitors, providing a faster Return on Investment (ROI) [O’Regan, 2017].

There are some international standards, frameworks, and models that define different programs for SPI, these are a set of best practices that are aligned with the business goals and play a key role in helping organizations to achieve their strategic goals. From all the models for SPI, the one that is more related to software development (practices for processes in software and systems engineering) is CMMI. CMMI contains a set of best practices from several areas, each of them to achieve a given purpose, that can include tools, methods, materials, and people. Organizations follow these practices to improve their software processes and meet their business needs more efficiently, as well as allowing them to do continuous improvement, adapt to technological innovations, and to prevent defects, in order to them to face future challenges [O’Regan, 2017, CMMI Product Team, 2010].
Currently, the CMMI framework addresses three constellations, that are known as collections of CMMI components for a specific area of interest. The focus of this research is in the development constellation also known as CMMI-DEV in the version 1.3 that is related to the development of both products and services [CMMI Product Team, 2010].

Due to the high complexity of CMMI-DEV in version 1.3 many organizations struggle to understand the model as they get lost in the various concepts and relationships in the model. To reduce the perceived complexity of the model, we propose to represent CMMI-DEV v1.3 with an EA using for that matter, the ArchiMate modeling language to represent both concepts and relationships of the framework and test if the reading of model in ArchiMate can be easier for the model users as well as analyze potential benefits of representing it with an EA.

1.1 Research Problem

The main benefits of adopting CMMI for process improvement are the reduction of the overall costs, project schedule (improvements in schedule predictability and reductions in the time required to do the work), quality improvement (reductions in the number of defects), customer satisfaction and ROI. These benefits have been proven by many case studies of organizations that adopted CMMI from around the world and from different areas [Goldenson and Gibson, 2003]. Despite this, only a little fraction of the software developing organizations adopt CMMI, so it is important to find why organizations do not adopt CMMI.

The authors from [Staples et al., 2007] studied why organizations do not adopt CMMI by collecting and analyzing two months of sales data from an Australian company that sells CMMI appraisals and improvement services. Their findings show that most organizations do not adopt CMMI because the CMMI program is expensive, requires time that many organizations do not have, organizations do not understand the benefits of using it and many organizations think they are too small to adopt it [this being a direct influence from not understanding the benefits of CMMI and resources constraints (budget and time)].

Two years later the authors from [Khurshid et al., 2009] replicated the previous study in another country, Malaysia, by using data collected from three consulting companies that sell a CMMI Level 2 program subsidized by the Malaysian government. This study supports the study conducted by [Staples et al., 2007] and identifies the same adoption problems as well as the problem of organizations having other priorities than process improvement.

Another study [Wu et al., 2005], now related to organizations who have already adopted CMMI, was done through a study conducted in China with the purpose of investigating the adoption of CMMI. The authors inquired most of the organizations who have been rated in a certain CMMI assessment to find the reasons, success factors, benefits and problems in the adoption of CMMI. Through the survey's
data, they identified the following problems: the organizations think that CMMI is an over-complex and dogmatic process, the costs are high when adopting it and that there is a lack of automated supporting tools for CMMI.

Some of the problems previously mentioned regarding the low adoption and the adoption of CMMI are directly related to its complexity and difficulty to be understood. The existing textual reference model of CMMI-DEV v1.3 [CMMI Product Team, 2010] is ambiguous, has a lot of technical definitions and extensive text, which make difficult for users to understand, implement and accept it [Islam and Zhou, 2011]. Users tend to get lost as there are a total of twenty-two process areas and near two hundred practices in the model with various relationships between them, increasing the effort of users in reading this representation of the reference model. This, allied to the fact that CMMI tells what to do and not how to do it for process improvement increases its complexity.

Summarizing, the problem that this research will tackle is the high perceived complexity of CMMI-DEV v1.3 by its users. To tackle this problem we are going to propose a different way to represent the CMMI-DEV v1.3 model, by using EA models with graphical elements of ArchiMate which usually are more appellative to people and easier to understand.

1.2 Research Methodology

Most of the researches done in the academic study of Information Systems (IS) are characterized by two paradigms: Behavioral Science (BS) and DS. The BS paradigm addresses research through the development and justification of existing theories that explain or predict human or organizational behavior to address identified business needs. The DS paradigm attempts to extend the boundaries of human and organizational capabilities by addressing research through the building and evaluation of new artifacts to solve identified business problems and needs, giving focus to the utility of the artifacts [Hevner et al., 2004, Goldkuhl, 2016].

This research fits in the DS paradigm, as we are going to try to solve our research problem through the development of models and instantiations of these models (new artifacts). To guide a DS research, we propose to use the DSRM. The DSRM guides our work through the development and evaluation of our various IS artifacts.

The DSRM is an iterative methodology that incorporates principles, practices, and procedures to carry out a DS research in IS area by creating and validating artifacts to address a research problem. These artifacts are broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems) [Hevner et al., 2004, Peffers et al., 2007].

In our research, the artifacts that we are going to create and evaluate are going to be models and constructs, by models we are referring to the reference model of CMMI and by constructs the mapping
we chose between the CMMI and ArchiMate.

The DSRM is divided into 6 phases: Problem identification and motivation, Define the objectives for a solution, Design, and development, Demonstration, Evaluation, and Communication. We are going to specify in a brief description what is the goal of each phase [Peffers et al., 2007] and for each phase, we adapt it to fulfill our research (Figure 1.1).

1. **Problem identification and motivation**: defines the specific research problem and justifies the value of a solution. With the problem identified and defined, it can be used to develop an artifact that is going to solve the research problem, and that is going to be the research solution;

2. **Define the objectives for a solution**: infers the objectives of a solution from the problem definition and knowledge of what is possible and feasible. This objective is inferred rationally from the problem specification and can be quantitative or qualitative;

3. **Design and development**: creates the artifact. In this phase, before creating the artifact it should be determined what is the desired artifact’s functionality and its architecture. A design research artifact can be conceptually designed as an object in which a research contribution is embedded in the design;

4. **Demonstration**: demonstrates the use of the artifact to solve one or more instances of the problem. To do the demonstration, it is required effective knowledge of how to use the artifact to solve
the problem. The demonstration could involve the use of experimentations, simulations, case studies, proofs, or other appropriate activities;

5. **Evaluation**: observe and measure how well the artifact supports a solution to the problem. This activity involves comparing the objectives of a solution defined in the design and development phase to actual observed results from use of the artifact in the demonstration phase;

6. **Communication**: communicate the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences such as practicing professionals, when appropriate.

The DSRM by being an iterative process lets us iterate many times through various phases and in each iteration obtain frequent and valuable feedback for the design process and incremental improvement of it.

### 1.3 Dissertation Outline

The structure of this research is strongly influenced by the DSRM, following the various phases of the methodology. Chapter 1 starts by introducing the research, giving detail to the research theme, the research problem, and motivation behind the research work (first phase of DSRM), chosen methodology and structure of the research. Chapter 2 does not follow a specific phase of the DSRM but is essential to our research, as introduces the fundamental concepts and other author’s researches regarding the scope of our own research. Chapter 3 follows the second and third phase of DSRM by defining the objectives of the proposed solution and the proposed solution. Chapter 4 follows the fourth phase of DSRM by demonstrating the proposed solution in practice. Chapter 5 follows the fifth phase of DSRM by evaluating the utility of the proposed solution. The final Chapter 6 follows the last phase of DSRM by describing how did we communicate our work to a specific audience as well as having the conclusions, contributions, and limitations of this research and future work that can be done on the topic of the research.
2. Related Work

In this chapter, we start by giving a theoretical background of the main topics related to our research and introduce some researches that are relevant to support our proposed solution. For the theoretical background, we start by introducing CMMI with a main focus on the Development constellation (CMMI-DEV) in the version 1.3 and in one of improvements paths used in CMMI for process improvement, next we introduce the EA area and the modeling language used in this research, ArchiMate. For the related work, we introduce proposals of ontologies for CMMI and researches that use the ArchiMate language to model other frameworks that have similar concepts to CMMI, like Control Objectives for Information and Related Technologies (COBIT) 5, Information Technology Infrastructure Library (ITIL) and some International Organization for Standardization (ISO)’s.

2.1 Capability Maturity Model Integration

CMMI was developed by a group of expert professionals from industry, government, and the SEI at Carnegie Mellon University and it is used to guide organizations to improve their processes and, consequently, improve software quality, time and costs of development and productivity by describing an evolutionary improvement path from ad hoc, immature processes to disciplined, mature processes with improved quality and effectiveness, telling organizations what to do to but not how to do it to achieve the desired business goals [Chrissis et al., 2011, Chaudhary and Chopra, 2017].

The CMMI Institute that is part of the Information Systems Audit and Control Association (ISACA) family and who currently administrates CMMI defines it as “a capability improvement model that can be adapted to solve any performance issue at any level of the organization in any industry. The Model provides guidelines and recommendations for helping your organization diagnose problems and improve performance. Used by over 5000 organizations in more than 70 countries all over the world, CMMI helps you identify and achieve measurable business goals.” ¹

The CMMI framework currently addresses three constellations, that are known as collections of CMMI components for a specific area of interest, and that can be used to construct models, training materials, and appraisals related documents. The Development constellations model also know as CMMI for Developers (CMMI-DEV) is most suitable for organizations that want to improve the development of products and services solutions, the Acquisition constellations model also known as CMMI for Acquisition (CMMI-ACQ) is most suitable for organizations who want to improve the large-scale acquisi-

¹Definition of CMMI from CMMI Institute: https://cmmiinstitute.zendesk.com/hc/en-us/articles/216947067
tion of products, services, and solutions and finally the Services constellation also known as CMMI for Service (CMMI-SVC) is most suitable for organizations who want to improve the delivery of services, like Information Technologies (IT) services, help desk, and others [O’Regan, 2017, CMMI Product Team, 2010].

CMMI can be also used for assessment, with the Standard CMMI Appraisal Method for Process Improvement more known as SCAMPI, that can be used for evaluating the maturity of the organizations (staged representation) and capability of the processes (continuous representation), the maturity levels enables organizations to improve a set of related processes and the capability levels enable organizations to improve an individual process area or group of process areas chosen by the organizations. In both approaches to reach a particular level, an organization must satisfy all the goals of the process areas or set of process areas that are selected for improvement [O’Regan, 2017, Chrissis et al., 2011].

2.1.1 CMMI for Development

As we said earlier, our research focuses on the version 1.3 of CMMI-DEV, that is used by organizations from different industries to cover activities for developing both products and services. CMMI-DEV v1.3 is composed of practices that cover the areas of project management, process management, systems engineering, hardware engineering, software engineering, and other supporting processes used in development and maintenance of projects, and is constituted by twenty-two process areas divided into four categories: Process Management, Engineering, Project Management, and Support.

A Process Area in CMMI is a group of related practices in a certain area, that when are implemented collectively, satisfies a set of goals considered important for making improvement in that area. Each process area contains components that are divided into three categories: Required Components, Expected Components, and Informative Components. Each type contains the following components [CMMI Product Team, 2010]:

- **Required Components**: Components that are essential to achieve process improvement.
  - **Specific Goals**: Describes the unique characteristics that must be present to satisfy the process area.
  - **Generic Goals**: Goal statement that applies to multiple process areas.

- **Expected Components**: Components that describe the activities that are important in achieving a required CMMI component.
  - **Specific Practices**: Describe the activities that are expected to result in the achievement of the specific goals of a process area.
  - **Generic Practices**: Practice that applies to multiple process areas.
Informative Components: Components that help model users understand CMMI required and expected components.

- **Purpose Statements**: Describes the purpose of the process area and practices.
- **Related Process Areas**: Lists references to related process areas and reflects the high-level relationships among the process areas.
- **Example Work Products**: Lists sample outputs from a specific practice.
- **Subpractices**: Detailed description that provides guidance for interpreting and implementing a specific or generic practice.
- **Generic Practice Elaborations**: Provide guidance on how the generic practices can be applied uniquely to process areas.

The only component that is essential for process improvement are the goals and for that reason, they are the only required component in the model. A process area is only satisfied by achieving/satisfying all the goals of that process area, that is, the goals support the process area purpose and each purpose supports improvement. To achieve a goal organization must be performing some practices, CMMI provides some practices that an organization might perform to satisfy each goal, but the organizations might have entirely different practices and might have a different number of practices and this is the reason why the practices are expected components and not required. The other components, informative, as its name suggests, give information to user models on how to understand the required and expected components of the model.

### 2.1.2 CMMI Staged Representation

Despite the improvement paths (representations) are not the focus of our work, nor do they appear in our proposed CMMI-DEV v1.3 reference model, we introduce the staged representation (maturity levels) as the organization that we choose to demonstrate our proposed solution is adopting the Maturity Level 3 of CMMI.

The staged representation in CMMI, better known as the maturity levels, is a way to represent how the CMMI model is organized around the path towards process improvement that an organization can follow. This representation organize the CMMI model with five maturity levels, that are a evolutionary stage that provide a roadmap for process improvements in an incremental manner to an organization and each level (except the first) consist of several process areas that mature an important subset of the organization processes, preparing it to move to the next maturity level. To achieve a certain maturity level, all the process areas belonging to that maturity level and all the process areas from lower maturity levels must be implemented and institutionalized (defined, documented, educated and used) in an organization [O’Regan, 2017, Chaudhary and Chopra, 2017, Chrissis et al., 2011].
Below is a brief description of the type of processes in each level:

1. **Initial** - Ad hoc processes that are unpredictable, poorly controlled and reactive leading to delay and over the budget of the completed work;

2. **Managed** - Characterized reactive processes for projects, to plan, perform, measure and control them;

3. **Defined** - Characterized proactive processes for the organization, to be used organization-wide across projects, programs, and portfolios;

4. **Quantitatively Managed** - Processes measured and controlled;

5. **Optimizing** - Focus on continuous process improvement.

The full list of all the process areas of CMMI-DEV v1.3 with the associated maturity level, category, number of specific goals and practices is in Table 2.1. It is important to note that the generic goals and practices are applied differently for each process area and there is a total of three generic goals and thirteen generic practices.

<table>
<thead>
<tr>
<th>Process Area</th>
<th>Category</th>
<th>Maturity Level</th>
<th>Specific Goals</th>
<th>Specific Practices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Causal Analysis and Resolution (CAR)</td>
<td>Support</td>
<td>5</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Configuration Management (CM)</td>
<td>Support</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Decision Analysis and Resolution (DAR)</td>
<td>Support</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Integrated Project Management (IPM)</td>
<td>Project Management</td>
<td>3</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Measurement and Analysis (MA)</td>
<td>Support</td>
<td>2</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Organizational Process Definition (OPD)</td>
<td>Process Management</td>
<td>3</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Organizational Process Focus (OPF)</td>
<td>Process Management</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Organizational Performance Management (OPM)</td>
<td>Process Management</td>
<td>5</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Organizational Process Performance (OPP)</td>
<td>Process Management</td>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Organizational Training (OT)</td>
<td>Process Management</td>
<td>3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Product Integration (PI)</td>
<td>Engineering</td>
<td>3</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Project Monitoring and Control (PMC)</td>
<td>Project Management</td>
<td>2</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Project Planning (PP)</td>
<td>Project Management</td>
<td>2</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>Process and Product Quality Assurance (PPQA)</td>
<td>Support</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Quantitative Project Management (QPM)</td>
<td>Project Management</td>
<td>4</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Requirements Development (RD)</td>
<td>Engineering</td>
<td>3</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Requirements Management (REGM)</td>
<td>Project Management</td>
<td>2</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Risk Management (RSKM)</td>
<td>Project Management</td>
<td>3</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Supplier Agreement Management (SAM)</td>
<td>Project Management</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Technical Solution (TS)</td>
<td>Engineering</td>
<td>3</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Validation (VAL)</td>
<td>Engineering</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Verification (VER)</td>
<td>Engineering</td>
<td>3</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

2Description of the processes in each level from CMMI Institute: https://cmmiinstitute.com/learning/appraisals/levels
2.2 Enterprise Architecture

Currently, to manage the complexity of any large organization or system, an architecture is needed. Architecture can be described as the properties and concepts that are fundamental to a system in their environment, incorporated into its elements, relationships and in the principles required for its analysis, design, and evolution [Lankhorst, 2017].

The use of the term architecture in the IS area started in the 80s, with the emerging of the ARIS framework in Europe, and in North America with the PRISM project and with a paper [Zachman, 1987] by Zachman introducing the Zachman Framework [Greefhorst and Proper, 2011].

In the start, from all of these proposals, the Zachman Framework [Zachman, 1987] was the only focused on the computerized IS, with the main objective of defining a mechanism to design architectures for representing organizations. The framework recognized two dimensions: one for the perspectives of specific target audiences, with the perspectives of different stakeholders, giving logical names for these perspectives (contextual, conceptual, logical, physical and out-of-context); and the other dimension for the types of artifacts, that answer the six organizational questions (what, how, where, who, when, why).

The result of these two dimensions gave form to the Zachman Framework [Zachman, 1987], that is a matrix with the different perspectives in the rows and the types of artifacts in the columns [Greefhorst and Proper, 2011]. The Zachman Framework is define as an ontology (metamodel) for describing the enterprise and not a methodology for creating the implementation (an instantiation) of the enterprise.

The use of architecture in an organization gave form to the EA term which is defined by Lankhorst [Lankhorst, 2017] 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".

Nowadays, business performance increasingly depends on a balanced and integrated EA, involving stakeholders, their competencies, organization structures, business processes, IT, finances, products, and services, as well as its environment. EA is seen as a holistic view of current and future representation of an organization that captures the essentials of business, IT, and its evolution and helps to design the components and relationships through the various layers of the architecture of an organization [Lankhorst, 2017, Greefhorst and Proper, 2011].

To build an EA, there are some frameworks and standards that guide architects through all phases of the life cycle of architectures [Lankhorst, 2017]. The Open Group Architecture Framework (TOGAF) is a framework well known for this matter. It offers mechanism and tools to build consistent EAs, that uses the best practices, reflects the needs of stakeholders and that gives attention to both the current requirements and the future needs of the business [The Open Group, 2018].

\[^{3}\text{Definition of Zachman Framework by Zachman https://www.zachman.com/about-the-zachman-framework}\]
One of the main components of the TOGAF is the Architecture Development Method (ADM), which is a generic and iterative method that describes a step-by-step approach of how to develop and manage the life cycle of an EA in face of its business needs and it’s aligned with the business strategy, the ADM forms the TOGAF core and have nine phases representing the TOGAF life cycle. Each of these phases feeds the overall requirements management process throughout the project lifecycle as seen in Figure 2.1 [Lankhorst, 2017, The Open Group, 2018].

![Figure 2.1: TOGAF Architecture Development Method.](image)

Another standard that is directly aligned to the TOGAF is the model language ArchiMate, that is used to build the required views for ADM, this language is divided into three main layers that are mapped to the phases of, business architecture (Phase B), information systems architectures (Phase C), and technology architectures (Phase D) seen in Figure 2.1 [The Open Group, 2018].

One of the advantages of using EA models to represent these types of SPI standards, frameworks and models is the fact that visual models are a more comprehensible, readable and understandable representation than a textual representation, making knowledge and information more explicit, lowering the user’s perceived complexity and therefore facilitates the learning of this type of frameworks [Almeida et al., 2017].
2.3 ArchiMate

ArchiMate is an EA modeling language developed and maintained by the same entity responsible for TOGAF, The Open Group, and the focus is mainly in enterprise modeling, allowing architects to describe, analyze and visualize business architectures, using for this purpose graphical concepts for the entities and relationships [The Open Group, 2017].

ArchiMate provides uniform representations of EAs by the form of models, that contain well-defined relationships allowing to connect concepts from different domains of the business architectures. In order to do this, it defines several layers in the EA (Figure 2.2), from which three main ones stand out, Business Layer, Application Layer, and Technology Layer, relating and distinguishing each of them with a service-oriented approach, that is, the layers expose their functionality as a form of services to the layers above [The Open Group, 2017].

A layer in ArchiMate is defined as an "abstraction of the ArchiMate framework at which an enterprise can be modeled" and in each layer there are three types of elements: the Passive and Active structure and Behavior, these aspects are defined in ArchiMate as "classification of elements based on layer independent characteristics related to the concerns of different stakeholders". The full framework of ArchiMate it is showed in Figure 2.2 with all layers and aspects [The Open Group, 2017].

Figure 2.2: Full ArchiMate Framework

The early versions of ArchiMate did not cover the concepts that motivated the design and operation of the EAs, that is, the "why" of the Zachman Framework, being necessary for the following versions to add an extension to the core of ArchiMate, that introduces motivation concepts, which allows modeling
the motivations or reasons underlying the design and changes in an EA. This extension is very important in our research because some of the concepts in IS frameworks represent the motivation of the organization or individual to achieve certain results, and CMMI is no different, there are objectives and desired properties to achieve the process improvement.

ArchiMate also defined a core set of generic relationships that can be used to connect the generic concepts and give a meaning to the way we look at the models. These relationships are “overload”, which means that the exact meaning of the connection between two concepts differs, depending on the source and destination concept [The Open Group, 2017].

2.4 CMMI Ontologies

There is no proposal on how to model CMMI in ArchiMate, but there are authors that propose other CMMI ontologies and metamodels using different modeling languages.

In the paper [Musat et al., 2010], following the paradigm Model Driven Development, the authors proposed a tool that supports the automatic generation of a language that can be used to specify practices of process areas. The automatic generation is performed through a CMMI metamodel in Unified Modeling Language that they propose as well and that can be seen in Figure 2.3.

![Figure 2.3: CMMI UML Ontology](image)

In the paper [Halit Soydan and M. Kokar, 2012], the authors propose a CMMI-DEV v1.3 ontology based on Web Ontology Language (OWL), a primary language for the Semantic Web. The authors
followed the same approach used in this language, first they formalize an ontology for the CMMI-DEV v1.3 that captures the main concepts and then they use a generic OWL reasoner to check the consistency of the representation of the CMMI-DEV v1.3 ontology to derive the classification of the level of maturity of the organization’s development process. This will allow to determinate the maturity levels of organizations through their data on the practices performed.

The authors from [Gazel et al., 2012] propose an ontology-based software process assessment tool with the objective of supporting the data collection phase of process assessment and to track compliance of software processes to CMMI. For this, the CMMI ontology (Figure 2.4) proposed was to create a base for mapping the organizational processes and CMMI-DEV v1.2 components for supporting process assessment activities and to reflect the CMMI domain knowledge as possible.

These researches were an important contribution to our work, as they will be used as a starting point for developing the proposed CMMI-DEV v1.3 metamodel in ArchiMate that contains the main concepts of CMMI-DEV v1.3 and their relationships.

2.5 Information Systems Frameworks and ArchiMate

The use of the ArchiMate modeling language to model frameworks related to the IS area and to enable the mapping with a standard based EA representation it is well documented. We will highlight

![Figure 2.4: CMMI Ontology with the Representations](image-url)
some of the researches conducted on this topic.

Lourinho et al. [Lourinho et al., 2017] propose in ArchiMate a metamodel for the ISO 27001 standard extended with the concepts of ISO TS 33052 and 33072, with the purpose of reducing the perceived complexity of implementing these IT Governance frameworks and map this metamodel with COBIT 5 metamodel proposed by Almeida et al. [Almeida et al., 2016] to show a complementary way of integrating both (Figure 2.5). The same COBIT 5 metamodel, is used in another research by Percheiro et al. [Percheiro et al., 2017], that propose a metamodel of ITIL in ArchiMate and an integration metamodel, integrating COBIT 5 with ITIL.

Figure 2.5: ISO 27001 and ISO TS 33052/33072 Metamodel

In other researchers, Vicente et al. [Vicente et al., 2013b, Vicente et al., 2013a], propose in ArchiMate a business motivation model for ITIL and a business-specific architecture using the principles of ITIL and the EA approach.

In other research, Gomes et al. [Gomes et al., 2017] propose adequate EA viewpoints to assist
Business Continuity Planning using one of COBIT 5 processes modeled in ArchiMate and demonstrating its utility through the mapping of a COBIT 5 process with an EA representation of a real organization.

And finally, Silva et al. [Silva et al., 2015] propose in ArchiMate a model for TIPA, a framework for assessing maturity for those who use ITIL and demonstrates how the model in ArchiMate allows alignment between service management and the organization's concepts and artifacts in a standardized way.

These researches were an important contribution to our work, as they will be used as a starting point when mapping ArchiMate with the CMMI, these frameworks have concepts that have similar meanings to the CMMI concepts, and the ArchiMate concepts that these authors chose to represent them can be used in our research. Another thing is the fact that the majority of these researches follow the DSRM and the way they demonstrate and validate their artifacts can be applied to our own research.
3. Research Proposal

This chapter corresponds to the "Define the objectives for a solution" and "Design and development" phases of DSRM [Hevner et al., 2004], where we explain our research proposal to solve the research problem as well as define the main objectives that the proposed solution must satisfy in order to address the research problem.

The purpose of the CMMI-DEV v1.3 reference model is to facilitate the learning of this framework, as well as understand what are the main components and their relationships. To solve the problem previously identified, we propose a metamodel of CMMI-DEV v1.3 developed in ArchiMate, containing the main components and relationships between them, and further ahead a model more detailed that we named CMMI-DEV v1.3 Reference Model. The metamodel has the main concepts of the CMMI-DEV v1.3 and their relationships, giving an overview with a focus on the metaclasses that represent them. The reference model is a more detailed model that has all the instantiations of the metaclasses identified in the metamodel and represent the full CMMI-DEV v1.3 framework using several views.

3.1 Objectives of the Proposal

To solve the problem identified and to follow the "Define the objectives for a solution" phase of DSRM [Hevner et al., 2004], the solution that we are going to propose has to be defined with focus on the following objectives:

- The solution must represent all the main concepts and relationships of CMMI-DEV v1.3 framework with an EA representation;
- The solution must allow users to navigate to any part of the CMMI-DEV v1.3 framework.

All of these objectives are important to achieve the solution to our problem and should all be achieved to solve the main objective of this research, that is to address the high perceived complexity of CMMI-DEV v1.3 by its users.

3.2 Mapping CMMI-DEV v1.3 with ArchiMate

To create our models in ArchiMate, we first chose which concepts of ArchiMate could represent the CMMI concepts. In order to do this, we did the mapping shown in Table 3.1 based on the previous studies.
in using ArchiMate with IS frameworks (Section 2.5). Then we identify the main relationships between the CMMI concepts and chose which ArchiMate relationships to represent them based on the textual reference model of CMMI [CMMI Product Team, 2010] and the previous studies of CMMI ontologies (Section 2.4).

<table>
<thead>
<tr>
<th>CMMI</th>
<th>ArchiMate</th>
<th>Justification [CMMI Product Team, 2010, The Open Group, 2017]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Area Category</td>
<td>Grouping</td>
<td>A Process Area Category in CMMI represents a group of Process Areas from the same area of interest and can be represented by the Grouping concept which in turn can represent an element that composes concepts that belong together based on some common characteristic.</td>
</tr>
<tr>
<td>Process Area</td>
<td>Grouping</td>
<td>A Process Area in CMMI is defined as a cluster of related practices and can be represented by the Grouping concept which in turn can represent an element that composes concepts that belong together based on some common characteristic.</td>
</tr>
<tr>
<td>Purpose</td>
<td>Goal</td>
<td>A Purpose statement in CMMI describes the purpose of the process area and can be represented by the Goal concept which in turn can represent a high-level statement of intent, direction, or desired. In this case, it is due to the purpose of CMMI represent the intention of satisfying the process area.</td>
</tr>
<tr>
<td>Specific Goal</td>
<td>Goal</td>
<td>A Specific Goal in CMMI describes the unique characteristics that must be present to satisfy the process area and can be represented by the Goal concept which in turn can represent a high-level statement of intent, direction, or desired. In this case, it is due to the fact that the goals represent the desired characteristic to implement a process area.</td>
</tr>
<tr>
<td>Generic Goal</td>
<td>Goal</td>
<td>A Generic Goal in CMMI is called “generic” because the same goal statement applies to multiple process areas, and can be represented by the Goal concept for the same reason as in the Specific Goal. A Specific Practice in CMMI is the description of an activity that is considered important in achieving the associated specific goal and can be represented by the Business Process concept which in turn can represent a human activity.</td>
</tr>
<tr>
<td>Specific Practice</td>
<td>Business Process</td>
<td>A Specific Practice in CMMI is the description of an activity that is considered important in achieving the associated specific goal and can be represented by the Business Process concept which in turn can represent a human activity.</td>
</tr>
<tr>
<td>Generic Practice</td>
<td>Business Process</td>
<td>A Generic Practice is called “generic” because the same practice applies to multiple process areas and can be represented by the Business Process concept for the same reason as in the Specific Practice.</td>
</tr>
<tr>
<td>Subpractice</td>
<td>Business Process</td>
<td>A Subpractice in CMMI is defined as a detailed description that provides guidance for interpreting and implementing a specific or generic practice and can be represented by the Business Process concept for the same reason as in the Specific and Generic Practice.</td>
</tr>
<tr>
<td>Example Work Products</td>
<td>Business Object</td>
<td>The Example Work Products in CMMI defines section lists of sample outputs from a specific practice and can be represented by the Business Object concept which in turn can represent information produced and consumed by an activity.</td>
</tr>
<tr>
<td>Generic Practice Elaborations</td>
<td>Deliverable</td>
<td>Generic Practice Elaborations appear after generic practices to provide guidance on how the generic practices can be applied uniquely to a process area and can be represented by the Deliverable concept, which in turn can represent a support concept.</td>
</tr>
</tbody>
</table>

For the relationships, we chose the composition relationship of ArchiMate to represent that a concept of CMMI is constituted by others CMMI concepts. We use the serving relationship when a concept of CMMI provides something to another concept of CMMI, the realization relationship is used when a concept of CMMI helps to achieve another CMMI concept, the access relationship is used when referring that a concept of CMMI accesses (create) another CMMI concept and finally, for the association relationship, we use it when there are CMMI concepts that have some type of association that cannot be represented by other ArchiMate relationship.
3.3 CMMI-DEV v1.3 Metamodel

Based on the ArchiMate concepts and relationships that we chose to represent CMMI-DEV v1.3, we proposed a metamodel, as shown in Fig. 3.1 which can be read as follows: a process area category is composed of a set of process areas, each process area has a purpose and each process area is composed of practices (specific and generic) that also have a purpose. To satisfy a process area all the goals (specific and generic) have to be achieved or satisfied, and this can be done by performing all the practices (specific and generic) related to each goal, the achievement of all the goals also support the purpose of the process area. To achieve the generic goals in a particular process area the generic practice elaborations are used to provide guidance on how the generic practices can be applied. The practices are composed by subpractices, in the case of the subpractices of specific practices to verify if it is being implemented, there are example work products that are sample outputs of these subpractices. Each process area can be supported by other process areas as well as each practice can be supported by other practices and process areas.

Figure 3.1: CMMI-DEV v1.3 Metamodel in ArchiMate
3.4 CMMI-DEV v1.3 Reference Model

The reference model shows all the instantiations of the metaclasses identified in the metamodel, this model shows all the concepts of CMMI-DEV v1.3 and their relationships in a more detailed way. In Fig. 3.2 we have a part of the reference model. As we can see in this part of the model, the first level of abstraction contains the process areas categories all of them except Engineering are divided by basic and advanced relationships and each of them contains a set of process areas [CMMI Product Team, 2010]. For dimension reasons, the complete model of this part of the reference model is in Figure. A.1 of the Appendix A.

Figure 3.2: Part of CMMI-DEV v1.3 Reference Model
To navigate to other parts of the reference model we have the views concepts, that are used as links to other parts of the model and if clicked on, will give a more detailed model of the concept they are referring to. For instance, if we click on the “View for Basic Project Management Relationships” we navigate to a detailed model of the interactions among the basic project management process areas and with the other process area categories as shown in the Fig. 3.3. From this part of the model, we can read as an example that the process area Project Monitoring and Control (PMC) provides “Corrective Actions” to the process area Supplier Agreement Management.

![Figure 3.3: Basic relations of Project Management process areas](image)

Returning to the first level of abstraction of our reference model, we can navigate to other parts of the model, for instance to navigate to a process area we click on the view related to that area, for example, if we click on the “View for REQM” it will show a detailed model of the process area Requirements Management (REQM) as it is shown in Fig. 3.4. This part of the model contains all of the concepts of the process area, the purpose of the area, all the practices that compose the area, divided in specific and generic as well as all the high-level relationships with related process areas. From this model we can see that the process area Risk Management provides the “Identification and Analysis of Risks” to this
process area. For each type of practices, we have a different view, to give in more detail the practices that are necessary to achieve the goals and satisfy the process area.

Figure 3.4: REQM process area model

For the specific practices of this process area, we have the “View for REQM Specific Practices” that allows us to navigate to a detailed model of the specific practices of this process area. This view is shown in Fig. 3.5 as well as the view for the specific practice 1.2 “Obtain Commitment to Requirements”. In the part of Fig. 3.5 containing all the specific practices we can see that the process area REQM is satisfied by achieving the specific goal “Manage Requirements”, that can be accomplished by performing the five specific practices that compose the area. Achieving this goal also supports the purpose of the REQM area. By clicking in the “View for REQM SP 1.2” we navigate to the detailed model of the specific practice “Obtain Commitment to Requirements” shown in Fig. 3.5, we can see in this model what is the goal that this specific practice helps achieve, the subpractices that compose this specific practice, the example of work products of each subpractice and the process areas that support this specific practice. So, we can see in the model for instance, that the subpractice “Assess the Impact of requirements on Existing Commitments” has as output the “Requirements Impact Assessments” as well as see that the process area PMC provides the “Monitor of Commitments” to this specific practice.
Returning to the REQM process area view (Fig. 3.4), we can navigate to the model that represents the application of the generic practices to this process area by clicking on the "View for REQM Generic Practices". A part of this view is shown in Fig. 3.6 as well as a view for the generic practice 2.1 "Establish an Organizational Policy". In the part of Fig. 3.6 containing all the generic practices we can see that this model is very similar to the model of the specific practices previously shown. There are goals that satisfy the process area and are achieved by performing practices associated with them. This model is identical to every process area as the generic practices are common to all process areas, the only differences can be seen in the model of each generic practice and are the generic practice elaborations. By clicking on the "View for REQM GP 2.1" we navigate to the detailed model of the generic practice "Establish an Organizational Policy" shown in Fig. 3.6, we can see that to achieve the generic goal "Institutionalize a
Managed Process” in the process area REQM, the specific practice “Establish an Organizational Policy” can be applied using the generic practice elaboration. For dimension reasons the full model of the “View for REQM Generic Practices” is in Figure of the Appendix A.

![Diagram of REQM generic practices and generic practice 2.1 models](image)

**Figure 3.6**: REQM generic practices and generic practice 2.1 models

The full CMMI-DEV v1.3 reference model has twenty-two process areas divided by the four process area categories. Then, we have a view for each process area, for each type of practices in a process area and for each practice in the process areas. In total the proposed reference model has more than 450 views.
4. Demonstration

This chapter corresponds to the “Demonstration” phase of DSRM [Hevner et al., 2004], where we demonstrate the utility of our proposed solution in solving one or more instances of the research problem. For this matter, we did a field study in a Portuguese organization and did the demonstration with the mappings between their EA and the proposed reference model for CMMI-DEV v1.3 to show the benefits of representing CMMI-DEV v1.3 with an EA.

The field study was conducted in one of the biggest technological Portuguese organizations. One of their departments is adopting CMMI, more precisely, trying to achieve Level 3 of maturity to become CMMI-DEV v1.3 Level 3 certified. To achieve this level of maturity they must satisfy all specific goals related to the process areas that compose this level of maturity and levels bellow this. In total, they must satisfy the specific and generic goals of twenty process areas from the various category. To satisfy each specific goal they must give response to all specific practices related to each specific goal. When we say give response to a specific practice, we are saying that, in the organization, there has to be artifacts (activities, procedures, etc) and affirmations (from interviews) that meet the specific practices.

First, we start by modeling the AS-IS state of the organization. This state has the processes, documents, and tools used in the organization which are part of the scope of CMMI project, that is, the state of the EA before the adoption of CMMI. Then, we did mappings between the specific practices of each process area from CMMI-DEV v1.3 reference model and the organization AS-IS state. These mappings show the parts of the AS-IS state that are giving response to the specific practices of each process area from the reference model. We can achieve these mappings by reading both, CMMI-DEV v1.3 process areas sections in [CMMI Product Team, 2010] and the organization documentation.

To exemplify one of these mappings we are going to show in Figure 4.1 a model with the parts of the AS-IS state from the organization software development area that are giving response to the specific practices of the Verification (VER) process area of CMMI-DEV v1.3. At the top of this model we have all the specific practices that compose the process area VER and in the rest of the model, the AS-IS state of the organization related to software development, the process, procedures, templates, and tools used. We then have the relationships we found between the VER process area and this state of the organization EA, that identify the activities and documents that give response to the VER specific practices. For instance, we can see in this mapping that three of the specific practices related with peer reviewing of the process area VER (SP 2.1-2.3) are not mapped with any part of the AS-IS state of the organization as well as see that the specific practices VER SP 1.1, 1.2, 3.1 and 3.2 are being satisfied
with activities of the Definition phase of the Iterative Software Development process.

**Figure 4.1:** Mapping of VER process area with AS-IS state

We use the association relationship to indicate that an element of the EA gives response to a specific practice. These relationships have different colors to facilitate the reading of this mapping, for each specific practice we have a different color to represent the relationships associated with it, for instance, we use the green color to specify the elements of the EA that give response to the specific practice VER SP 1.2.

To see in detail what are the activities in the Definition phase of the Iterative Software Development process that give response to the specific practices VER SP 1.1, 1.2, 3.1 and 3.2, the model contains the view elements, that lets us navigate to a more specific model for each element of the model specifying in more detail the relationships between the element and the specific practices. For instance, to navigate to a more detailed model of the Definition phase from the Iterative Software Development process we click in the “View for Definition phase V3.0 VER” that is shown in Figure 4.2. This model contains the
activities of the phase, the roles involved in the activities, and the documents used in the activities which are represented with concepts from the Business Layer of Archimate and the tools used in the activities which are represented with concepts of the Application Layer of ArchiMare mapped with the specific practices. We can see as an example that the specific practice VER SP 3.1 and 3.2 are satisfied by the activity of "Control the Quality of deliverables".

Figure 4.2: Mapping of VER process area with Definition phase of the Iterative Software Development process in the AS-IS state

With this kind of mappings, we can represent in a different way a gap analysis, where we compare the actual state of the organization with the desired state in terms of an EA being CMMI compliant, identifying the strengths and improvement potentials regarding CMMI. In other words, we identify what are the specific practices from CMMI-DEV v1.3 that are not being satisfied in the current state of the organization (AS-IS), for the organization to know what are the practices they should focus on when developing and changing their EA to be CMMI compliant. These mappings also show what parts of the existing EA can be reused when developing the new EA of the organization compliant with CMMI-DEV v1.3, that is the TO-BE state.

For the other part of our demonstration, we used the same organization, but now they are already
in a stage where they changed their EA to be CMMI compliant, that is, they change their EA to give response to all the specific practices of all the process areas from Level 3 of maturity. Using the same approach as we did with the mappings between the AS-IS state and process areas of CMMI-DEV v1.3, we started by modeling the TO-BE state of the organization and then mapped with the specific practices of CMMI-DEV v1.3 process areas from the reference model. To compare the previously mapping we used again the VER process area and software development area of the organization. In Figure 4.3 we have the mapping between this process area and the TO-BE state of the organization.

Figure 4.3: Mapping of VER process area with TO-BE state

Comparing to the mapping between the AS-IS state and VER process area, we can see for instance that now the Definition phase of the Iterative Software Development process gives response not only to the specific practices VER SP 1.1, 1.2, 3.1 and 3.2 but also the SP 2.1, 2.2 and 2.3. Again, if we want to
see in more detail what were the changes in this phase to give response to new specific practices, we click on the view element related to this phase. We can see in Figure 4.4 the model associated with this view. In this model, we see can now that the activity of "Control the Quality of Deliverables" calls a new procedure "Peer Reviews" that give response to the specific practices VER SP 2.1, 2.2 and 2.3 making the activity itself to give response to the same specific practices.

![Figure 4.4: Mapping of VER process area with Definition phase of the Iterative Software Development process in the TO-BE state](image)

With this kind of mappings, we can check if the new EA was developed to be compliant with CMMI-DEV v1.3. To see if the EA was correctly developed, all the specific practices of the process areas of CMMI-DEV v1.3 have to be accomplished (mapped with the EA), buy looking again to Figure 4.3 we can see that all the specific practices of the process area VER are mapped with the TO-BE state of the organization.

Having both EA states of the organization, we then color them to emphasize what were the changes in the organization to be compliant with CMMI-DEV v1.3. We use the colors Red, green and blue. Red for parts of the EA that have been removed (presented in AS-IS but not in TO-BE), green for parts of the EA that are new (presented in TO-BE but not in AS-IS) and blue for parts of the EA that were reused.
(presented in AS-IS and TO-BE). In Figure 4.5 we can see the differences between the phase Definition from the Iterative Software Development in the state AS-IS (top of the figure) and TO-BE (bottom of the figure) with the use of the colors. We use different tones for each color when a concept is overlapping other (composition) to facilitate the reading of these models.

![Figure 4.5: The two states of the EA with the proposed colors](image)

By coloring both the AS-IS and TO-BE state of the organization we show them a different way to see the changes they made to their EA while adopting CMMI-DEV v1.3. Giving an idea of what were the changes and by locking to the mappings between the TO-BE state and the process areas of the
reference model the organization can see the purpose of the change, that is, what specific practices do the new parts of EA give response to. This can also be useful for the organization for futures migrations of CMMI, as they can check what were the changes and why the changes to be compliant with CMMI-DEV v1.3 and see this as a blueprint for the adoption of CMMI-DEV v1.3, and further ahead use this knowledge for future migrations of CMMI.
5. Evaluation

This chapter corresponds to the “Evaluation” phase of DSRM [Hevner et al., 2004], where we evaluate the utility of our proposed solution to tackle the defined research problem of the high perceived complexity of CMMI-DEV v1.3. First, we start by using known criteria and methods used in the evaluation of Design Science Research (DSR) artifacts to evaluate the mapping we did between ArchiMate and CMMI (construct) and to evaluate the quality of the CMMI-DEV v1.3 reference model (model). Afterward, with the use of the demonstration in a real organization, we did interviews with open discussions with relevant stakeholders in the organization to validate both the utility of the artifact and the demonstration itself.

5.1 Artifact Evaluation in DSR

To support the evaluation of our proposed DSR artifacts, we will use the criteria identified by Prat et al. [Prat et al., 2015] that identified the main criteria for artifact evaluation used in researches that follow the DSRM.

These authors concluded that DSR artifacts can be viewed as systems and divided the identified criteria by five systems dimensions: Goal; Environment; Structure; Activity and Evolution. The ones we found most relevant belong to the structure dimension that is more related to the evaluation of models and constructs. We chose the following criteria:

- **Completeness**: The degree to which the structure of the artifact contains all necessary concepts and relationships between concepts;
- **Simplicity**: The degree to which the structure of the artifact contains the minimum number of concepts and relationships between concepts;
- **Style**: The elegance with which the artifact has been built;
- **Homomorphism** (Correspondence with another model): The degree to which the structure of the artifact corresponds to a reference model.

For Completeness, we can say that the mapping we did between CMMI and ArchiMate is complete, as every CMMI concepts were mapped to an ArchiMate concept, which means that CMMI-DEV v1.3 can be completely represented using ArchiMate.
As for **Simplicity**, with the use of the view concepts, we can represent each concept of CMMI with their own model, avoiding that the CMMI-DEV v1.3 reference model is only one big complex model but a model composed of less complex models and with fewer concepts and relationships.

For **Style**, some choices have been made for a better reading of the CMMI-DEV v1.3 reference model as for instance: We avoid the overlap of lines representing the relationships; We used the same color for concepts of the same type and layer of architecture (e.g., yellow for concepts of the business layer) and changed the tone of the color when a concept is overlapping other (composition); As for the textual description of all concepts to be visible, we resize the graphic concepts of ArchiMate; For the names of the concepts, the views start always with “View for”, for the Specific and Generic Goals and Practices the names have the sequential number given in [CMMI Product Team, 2010], in the case of the specific goal and practice the name starts with the prefix of the process area (e.g., REQM SG 1 for the first specific goal of the process area REQM and REQM SP 1.1 for the first specific practice associated with the first specific goal of the process area REQM).

As for **Homomorphism**, it can be evaluated with the support of the Bunge-Wand-Weber ontological analysis method [Wand and Weber, 1993], that let us compare two grammars and identify ontology deficiencies between them. There are four deficiencies that the method states (Fig. 5.1):

![Ontological deficiencies](image)

*Figure 5.1: Ontological deficiencies [Wand and Weber, 1993].*
– **Incompleteness**: can each concept of the set from the first grammar be mapped with a concept from the set of the second grammar? (the mapping is partial if it’s not total);

– **Redundancy**: can each concept of the set from the first grammar be mapped with more than a single concept from the set of the second grammar? (the mapping is redundant if it is not ambiguous);

– **Excess**: can each concept of the set from the second grammar be mapped with a concept from the set of the first grammar? (the mapping is excessive if there are concepts of the second grammar without a mapping);

– **Overload**: can each concept of the set from the second grammar be mapped to exactly one or more concept from the set of the first grammar? (the mapping is overloaded if at least one concept from the set of the second grammar is mapped to more than one concepts in the set of the first grammar).

Having the definitions of each deficiency, we can evaluate the mapping between CMMI (first grammar) and ArchiMate (second grammar). We have already see that CMMI can be completely represented with ArchiMate, so we did not find **Incompleteness** in our mapping.

As for **Redundancy**, we did not find it because there was no concept from CMMI that could be represented by more than one ArchiMate concept. This may be a consequence of the CMMI concepts being well defined and not ambiguous in their description.

We found **Excess** because we can not map all of the concepts of ArchiMate with CMMI, there are many elements from ArchiMate that are not mentioned in CMMI. Being a framework of best practices, CMMI describes activities and motivations for process improvements and it’s normal that does not mention concepts related to the application and technology layer of ArchiMate for instance.

Finally, we found **Overload**. There are concepts of ArchiMate that represent more than one concept in CMMI. For instance, the business process of ArchiMate can represent the specific practice, generic practice and subpractice of CMMI and the goal of ArchiMate can represent the goals and purpose in CMMI. This can be a problem if we want to do the opposite process of going from the CMMI-DEV v1.3 reference model in ArchiMate back to textual reference model of CMMI-DEV v1.3.

From all the criteria used in this critical evaluation, the only one that was not satisfied was **Homomorphism**, we found instances of the two deficiencies **Excess** and **Overload**. **Excess** does not present a real problem and **Overload** if occurs can be fixed by adding properties to the ArchiMate concepts. While modeling, we can add a property to the ArchiMate object with the type of CMMI concept that the ArchiMate concept is representing.
5.2 Interviews

Interviews done to relevant stakeholders can be used as suitable data generation method to validate and get feedback about the utility of the proposed solution and demonstration. From the interviews, we can get depth information about the utility of our models from the various stakeholders interested in this theme (CMMI professionals, professors and staff from organizations that are adopting or adopted CMMI). This happens because there’s more interaction with the stakeholders and some topics arise during the interviews as well ask follow-up questions [Turner, 2010].

Therefore to validate our proposal, we interviewed five IT people from the organization we use to demonstrate our proposed solution, these people were part of the team responsible to adopt/implement CMMI-DEV v1.3 Level 3 of maturity in one of the departments in the organization and had formation in this version of CMMI, so they are familiar with CMMI and its terms.

Before the interviews, we presented our work to them, showing them our proposed CMMI-DEV v1.3 reference model and demonstrating to them the utility of representing CMMI with an EA, by showing the mappings with did between the reference model and their EA before (AS-IS) and after the adoption of CMMI (TO-BE). After the presentation, we ask them some questions that had the following general opinions and some testimony from the interviewees that can be read in the Table 5.1.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Opinions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you think that a model with graphical elements can be useful for anyone who is learning or using CMMI?</td>
<td>The interviewees said that without a doubt, it is a great advantage as an alternative to reading a book, for them, the model shows the relationships that are only detailed in CMMI formations that many times are used to help formants to learn on how to read the CMMI book and where to look for information.</td>
</tr>
<tr>
<td>Do you think that the CMMI-DEV v1.3 reference model presented could facilitate the use of CMMI?</td>
<td>The interviewees said that without a doubt, as they said during the interview the most difficult part do understand in the CMMI model is the relationships between the practices and process areas and with this reference model they can identify them easier.</td>
</tr>
<tr>
<td>The mapping shown between your organization’s EA (AS-IS) and the CMMI-DEV v1.3 reference model facilitates the representation of a gap analysis?</td>
<td>Testimony: “From a general point of view and not only applied to this case, I am always a defender that a picture is worth more than a thousand words, so answering the question between the text and the diagram, I think that the graphic representation is always much more explanatory than to having a text explain the differences”. The other interviewees focus the fact that they did the gap analysis in an Excel sheet where they put a color scheme to realize which practices they fulfill and those that they do not fulfill, and with this, they did not perceive and visualize the practices in the whole organization as opposed to the representation of the gap analysis with these mappings.</td>
</tr>
<tr>
<td>The two mappings shown between your organization’s EA (AS-IS and TO-BE) and the CMMI-DEV v1.3 reference model allowed you to check the changes that occurred and why these changes in the adoption of CMMI more clearly?</td>
<td>Testimony: “Indeed, we can check what were the changes in an easier way with the colors, the only problem that I can see in the colors you chose is in the text of the colored parts in blue, they should be in white instead of black to a better reading of the models”.</td>
</tr>
<tr>
<td>Can these mappings be important for future CMMI migrations?</td>
<td>Testimony: “Yes, I think that for the adoption of the new version would be super useful, by the way, these models should be part of some walls in here for us to go there and complete the comparison with the new version, at least would be interesting”.</td>
</tr>
</tbody>
</table>
Unanimous the opinions of the interviewees were positive. For them, the new CMMI-DEV v1.3 reference model is a good way to represent this framework to users in general. They also, stressed out that the proposed representation of CMMI with an EA can be very useful because of the mappings that can be done between this representation and the EA of an organization. Mappings, that can represent the steps of an organization when adopting CMMI (gap analysis, process and practices mapping, etc) and be used for formations in the organization and as CMMI adoption history that can be used in future migrations.
6. Conclusion

This research was developed with the purpose of solving the problem of the high perceived complexity that users have when reading the textual representation of the CMMI model, with a focus on the version 1.3 of CMMI-DEV. There are already a few studies that approached some problems related to other IS frameworks using EA and ArchiMate to tackle them but there is little research about EA and ArchiMate with CMMI.

Thus, to solve this problem, we chose to use an EA approach and develop a reference model using the modeling language ArchiMate for the CMMI-DEV v1.3. With the visual representation of this reference model and based on the interviews we did, we believe that we can lower the user’s perceived complexity of CMMI and therefore contribute to turning the CMMI framework easier to use, allowing users to read and understand the CMMI framework more easily and in a more interactive way.

Also, we can take advantage of using the EA approach of representing CMMI with the ArchiMate modeling language, for instance, we can represent the assessment in an organization with models, showing them a different way to see a gap analysis by mapping CMMI practices of the reference model with the present state of an organization (AS-IS) and show what parts of the organization EA give response to CMMI and what practices are not satisfied.

We also can do mappings between the CMMI-DEV v1.3 reference model and the EA of an organization who already adopted CMMI to verify if their EA is compliant with this version of CMMI, that is, all the specific practices of CMMI are being mapped with the EA of the organization.

Having both mappings, we can colored both states of an organization to identify what where the changes in the EA of the organization after adopting CMMI (AS-IS vs TO-BE), underlying the changes they had to make to be CMMI compliant and with the mappings, identify the causes of the changes (practices that are now satisfied by the referred change).

6.1 Limitations

There were some limitations during our research. In the development of the models presented in this research, the tool used was the Archi modeling toolkit, that is an opens source modeling tool for ArchiMate. This tool has many limitations like the fact that is not scalable and do not allow cross model-dependencies that implying that all our models need to be in the same model package to link them. For instance to do the mappings shown in this research, both CMMI-DEV v1.3 reference model and the EA of the organization used to demonstrate the utility of the reference model had to be in the same model
package making it even harder to manage the many entities and relationships inside the package.

Another limitation of this research was the constant delays in the launch of the new version of CMMI which imply that in this research we did not focus on it as planned at the beginning of the research. The new version was expected to be launch in last year but was postponed to end of March of this year.

In terms of limitations of the proposed work, there are some that are important to be referred:

- The mappings done show that a document can satisfy a set of practices but does not show what is the part of the document that satisfies each practice;
- The mappings done show that an activity in AS-IS does not respond to a specific practice but in the TO-BE the same activity already gives an answer to a specific practice, the mapping does not show what has been changed in the content of the activity for this to happen;
- The mappings done show the EA of an organization from their documentation, which may not represent the actual EA;
- The mappings done cannot represent if a specific practice is partially satisfied (parts of EA give response to a specific practice but not fully).

6.2 Research Communication

This section follows the last phase of DSRM, the "Communication" phase [Hevner et al., 2004], where we describe how did we communicate our research to relevant audiences, showing them the importance of the defined problem and the utility and effectiveness of the proposed artifacts in tackling it.

This document itself is part of the communication of our research, we are going to present, discuss and debate it with a qualified jury that is going to evaluate our work to later become public to everyone.

Another way to communicate our work was to submit an abstract that was accepted to the Capability Counts 2018, a conference organized by the CMMI Institute which took place on May 1 and 2 of 2018 in Reston, Virginia, USA. In this conference we did a presentation, where we showed our research to the professionals of CMMI and to the people from CMMI Institute responsible for administrating CMMI that became very interested in our work and ask us to send the reference model that we propose and to future collaborations.

Finally, we publish a paper in the 26th International Conference on COOPERATIVE INFORMATION SYSTEMS (COOPIS 2018) (Rank A), we submit a research in progress with the title CMMI-DEV v1.3 Reference Model in ArchiMate. In this paper, we still did not show some parts of the demonstration as well as the part of the evaluation with interviews we already showed in this document and for that matter,

---

1Capability Counts 2018: http://cmmiinstitute.com/conferences
2COOPIS: http://www.otmconferences.org/index.php/conferences/coopis18
it was classified as research in progress. This conference is going to take place in Valletta, Malta on October 22-26.

6.3 Future Work

For future work, several things could be done. In the limitations section, we mention the Archi tool. So furthermore, to show these type of models and mappings between models the use of another tool would be a good idea like the use of the BiZZdesign Enterprise Studio\(^3\) or EAMS\(^4\) that would facilitate the development of our artifacts and the demonstration and evaluation of them.

Also, as future work, an interesting thing would be to apply the same approach to the new version of CMMI, the version 2.0. First map the new concepts of CMMI-DEV v2.0 with ArchiMate concepts and then develop a reference model for this version. Having both references models we then could do a mapping between them and show what were the changes as well as see what parts of the CMMI-DEV v1.3 maps to the CMMI-DEV v2.0. This can important for organizations that have already adopted the version v1.3 and have the mapping of their EA (TO-BE) with the v1.3 and by knowing with parts of the v1.3 maps with v2.0 they can do in an easier and faster way the mapping of their EA (AS-IS before adopting the new version) with new version of CMMI.

\(^3\)BiZZdesign Enterprise Studio: https://www.bizzdesign.com/enterprise-studio
\(^4\)EAMS: http://www.linkconsulting.com/eams/
Bibliography


A. Appendix - Parts of CMMI-DEV v1.3 Reference Model

Figure A.1: CMMI-DEV v1.3 Reference Model