

# A Quality Instrument for the Enterprise Architecture Development Process

Bas van den Bent

Master's thesis  
Information Science

Utrecht University & Sogeti Netherlands B.V.

INF/SCR-05-27

January 2006



**Author**

Bas van den Bent  
Utrecht University  
Institute of Information and Computing Sciences  
Master Business Informatics  
Student number: 0056219  
bbent@cs.uu.nl

**Utrecht University****Universiteit Utrecht**

Utrecht University  
Institute of Information and Computing Sciences  
Master Business Informatics  
<http://www.uu.nl>

Dr. Rik Bos  
[rik@cs.uu.nl](mailto:rik@cs.uu.nl)

Prof. dr. Sjaak Brinkkemper  
S.Brinkkemper@cs.uu.nl

**Sogeti Netherlands B.V.**

Sogeti Nederland B.V.  
Wildenborch 3,  
1112 XB Diemen  
<http://www.sogeti.nl>

Drs. Ir. Marlies van Steenbergen  
[marlies.van.steenbergen@sogeti.nl](mailto:marlies.van.steenbergen@sogeti.nl)

Drs. Martin van den Berg  
[martin.vanden.berg@sogeti.nl](mailto:martin.vanden.berg@sogeti.nl)



## **Abstract**

Up until this moment in time contributions to the field of enterprise architecture have been based on best-practice experiences. But while the field matures, the demand for scientific contributions to this field is growing. This thesis document describes a scientific contribution to the field of enterprise architecture in the form of an instrument for measuring the quality of the enterprise architecture development process. It describes the steps undertaken in the development and validation of this quality instrument.

First an extensive literature study is described. This study resulted in definitions of the central concepts used in this thesis. With these definitions in place, the second part of this literature study focuses on the theoretical framework that forms the foundation for the quality instrument. This theoretical framework combines the two main concepts, a quality model consisting of thirty-three quality attributes subdivided over seven dimensions and process viewpoints, into viewpoint representations that address different perceptions on the quality of the enterprise architecture development process. Based on this framework the quality instrument is developed. First relevant quality issues in the form of quality attributes are identified for every viewpoint representation. These issues are then transformed into items for the quality instrument. An instrument item is a statement relating to one development process quality sub issue that expresses the preferred state of the development process for that issue. The final step in the development of the instrument consists of adding a customized Likert scale to the items, on which respondents can rate their own enterprise architecture development process. Subsequently this document describes the successful validation of the resulting quality instrument by using cross-sectional data analysis and addresses the application of the instrument in practice.

This thesis project has resulted in a reliable and valid instrument for measuring the quality of the enterprise architecture development process. As such it provides both professionals and scientists with a valuable tool for determining the quality of architecture development processes. And although there remain many issues and opportunities for future research, this thesis project represents a valuable exploratory study into the quality of enterprise architecture and enterprise architecture processes.



## **Preface**

During the last two years of my doctorate Information Science at Utrecht University, I was introduced to the field of enterprise architecture. The combination of business and IT, the central concepts in this field, immediately appealed to me. Sogeti Netherlands B.V. is one of the leading companies in the field of enterprise architecture in the Netherlands. Their approach to enterprise architecture named DYA<sup>®</sup> is becoming the de-facto standard in the Netherlands. So when the opportunity presented itself to conduct my thesis project at Sogeti, I jumped at the opportunity.

I would like to thank Sogeti for providing the possibility and resources for conducting this research project. Special thanks go out to Marlies van Steenberg, my primary supervisor at Sogeti, for her guidance and advice. Her enthusiasm and sharp eye have been a great support during the eight months of my research. My thanks also go out to Martin van den Berg, my second supervisor, for his time and support. Last I would like to thank all architects and colleagues at Sogeti who were willing to dedicate some of their time to the interviews, the workshop and the questionnaires.

My supervisors at Utrecht University have played an important role in warranting the scientific contribution of this research. I thank Rik Bos and Sjaak Brinkkemper for their support and steering and for the opportunity to perform this research.

On a personal note, I would like to thank my parents for their support during my years at university. I thank Richt for proofreading my thesis. And I would also like to thank my friends and roommate for their support.

Finally, special thanks go out to my fellow student Elise together with whom I have spent my eight months at Sogeti. Sharing a room at Sogeti, she was a valuable source of support and motivation. I have learned a lot from working with her.

Utrecht, January 2006

Bas van den Bent



## Table of contents

Abstract .....	v
Preface .....	vii
Table of contents.....	ix
List of figures.....	xiii
List of tables.....	xiii

1 Introduction .....	2
1.1 Problem definition .....	2
1.2 Research methods.....	3
1.3 Previous research.....	4
1.4 Scientific & social relevance .....	5
1.5 Results .....	5
1.6 Thesis outline .....	5

## Part I: Theoretical Framework

2 Terminology.....	8
2.1 Introduction .....	8
2.2 Enterprise architecture.....	9
2.3 Enterprise architecture process .....	11
2.3.1 Development process .....	13
2.3.2 Maintenance process.....	13
2.3.3 Governance process.....	13
2.3.4 Deployment process.....	13
2.4 Enterprise architecture products .....	13
2.5 Quality .....	14
2.5.1 Quality attributes.....	14
2.5.2 Quality of the enterprise architecture process .....	14
2.5.3 Quality of the enterprise architecture products.....	15
2.6 Viewpoints .....	15
2.6.1 Viewpoint representation .....	16
3 Scope .....	18
4 Quality model.....	20
4.1 Introduction .....	20
4.2 Quality model I.....	20
4.2.1 Quality dimensions .....	20
4.2.2 Quality attributes.....	21
4.3 Quality model II.....	26
4.3.1 Quality attributes.....	26
4.4 Quality model III.....	27
4.4.1 Quality attributes.....	27
4.5 Summary.....	28
5 Viewpoints.....	30
5.1 Introduction .....	30
5.2 Viewpoints .....	30
5.2.1 Customer .....	30
5.2.2 Executor.....	30
5.2.3 Maintainer .....	30
5.2.4 Governor.....	31
5.2.5 Owner.....	31
5.3 Concerns .....	31
5.3.1 Questionnaire.....	31
5.3.2 Interactive workshop.....	33
5.3.3 Results.....	33

6	Framework .....	36
6.1	Introduction .....	36
6.2	Customer .....	36
6.3	Executor .....	37
6.4	Maintainer.....	37
6.5	Governor .....	38
6.6	Owner .....	38
6.7	Completeness.....	39

**Part II: Quality Instrument**

7	Construction .....	42
7.1	Introduction .....	42
7.2	Sub issues .....	42
7.3	Items .....	43
7.4	Scales .....	45
8	Validation .....	46
8.1	Introduction .....	46
8.2	Population.....	46
8.3	Statistical analysis.....	47
8.3.1	Reliability .....	47
8.3.1.1	Results .....	48
8.3.2	Validity .....	50
8.3.3	Results.....	51
8.4	Summary.....	53
8.4.1	The instrument.....	53
8.4.2	Gathered insights .....	53
9	Application.....	56
9.1	Introduction .....	56
9.2	Type of instrument.....	56
9.3	Application .....	56
9.4	Results .....	57

**Part III: Conclusions and Discussion**

10	Conclusions .....	62
11	Discussion .....	64
12	References .....	66
13	Appendices.....	69
	Appendix I: Definition of the Quality Attributes.....	69
	Appendix II: Definitions of the Viewpoints on the Development Process .....	72
	Appendix III: The Quality Instrument.....	73





## **List of figures**

- Figure 1: Thesis work plan and research methods
- Figure 2: A graphical representation of the important concepts used in this thesis.
- Figure 3: The sub processes of the enterprise architecture process
- Figure 4: The focus of the theoretical framework
- Figure 5: the quality model consisting of 33 attributes divided over 7 dimensions
- Figure 6: Quality attributes for the Customer viewpoint
- Figure 7: Quality attributes for the Executor viewpoint
- Figure 8: Quality attributes for the Maintainer viewpoint
- Figure 9: Quality attributes for the Governor viewpoint
- Figure 10: Quality attributes for the Owner viewpoint
- Figure 11: Benchmark results of all respondents over all attributes
- Figure 12: Example of the attributes completeness and consistency

## **List of tables**

- Table 1: Examined enterprise architecture approaches
- Table 2: Quality attributes added to the dimension Usability from QM I
- Table 3: Quality attributes added to the dimension Functionality from QM I
- Table 4: Quality attributes added to the dimension Efficiency from QM I
- Table 5: Quality attributes added to the dimension Scalability from QM I
- Table 6: Quality attributes added to the dimension Visibility & Control from QM I
- Table 7: Quality attributes added to the dimension Reliability from QM I.
- Table 8: Quality attributes added to the dimension Maintainability from QM I.
- Table 9: Quality attributes added to the dimension Functionality from QM II.
- Table 10: Quality attribute added to the dimension Usability from QM III.
- Table 11: Results of the first questionnaire
- Table 12: Results of the interactive workshop
- Table 13: The attribute completeness and its items
- Table 14: Population details (2004)
- Table 15: Cronbach's alpha acceptance levels
- Table 16: Results of the reliability study
- Table 17: Results of the factor analysis
- Table 18: Summary of the theoretical framework



# 1 INTRODUCTION

---

At the present time more and more organizations in the Netherlands are using enterprise architecture as a means for dealing with constant change within their organizations in a coherent and consistent way [44]. Approaches towards enterprise architecture such as DYA<sup>®</sup> [5, 45] in the Netherlands, are becoming more and more established. And although there still remains discussion about the concepts used, the field of enterprise architecture is maturing. At this moment in time the available approaches towards enterprise architecture are best-practice experiences bundled together. But as the field matures, the demand for the demand for scientific contributions grows.

In cooperation with Sogeti Netherlands B.V. this thesis project was aimed at realizing a scientific contribution to the field of enterprise architecture. Sogeti Netherlands B.V. is an IT service provider situated in the Netherlands. The core business of Sogeti is the development, testing and implementation of software applications. Next to its core business Sogeti also offers services in the field of process management and enterprise architecture. Currently Sogeti is the thought and market leader in software testing with their testing approach Tmap<sup>®</sup>. Sogeti has the ambition to achieve the same status with its approach to enterprise architecture, known as DYA<sup>®</sup>. Dynamic Architecture or DYA<sup>®</sup> is an example of an approach towards enterprise architecture that has been developed based on best-practices. DYA<sup>®</sup> provides enterprises with a framework that can help organize the people, products and processes involved in working under enterprise architecture. The combined ambitions of Sogeti and Utrecht University in the field of enterprise architecture have resulted in the opportunity to conduct this thesis project.

## 1.1 Problem definition

An in depth study into the field of enterprise architecture, its concepts and existing approaches provided numerous opportunities for scientific contributions. After dialogues with both the supervisors from Utrecht University and Sogeti, the quality of enterprise architecture was chosen as research subject.

However, the field of enterprise architecture has many different topics associated with it. The working definition for the concept enterprise architecture explained in section 2.2 speaks of products. These products are the results of, and input for a number of different processes, processes that are executed by people. Addressing this whole diversity of topics was not a realistic option, so the choice was made to narrow the scope of this research to the process of enterprise architecture. In a parallel research project Veltman [44] researched the subject of enterprise architecture products.

To guide the thesis project the main research question was formulated as:

“How can a scientifically validated method for measuring the quality of the process of enterprise architecture be created?”

To structure this thesis project, the main research question was decomposed into six sub questions. These questions were formulated as follows:

1. What does a generic process of enterprise architecture look like, in which phases can this process be decomposed?
2. What defines the quality of the process of architecture development?
3. How can quality attributes, as used in other fields such as the development of software products, be used in determining this quality?

4. How can these quality attributes be mapped to the generic process of working under architecture?
5. How can the quality attributes of the enterprise architecture process be measured, in other words what are the scales and limits of the defined attributes?
6. How should the process of measuring this quality be set up?

## 1.2 Research methods

During this thesis project a number of different research methods were employed. Figure 1 shows the subsequent steps that were taken during this thesis project. It also shows the research methods that were used in each of the steps. These methods are commented on below.

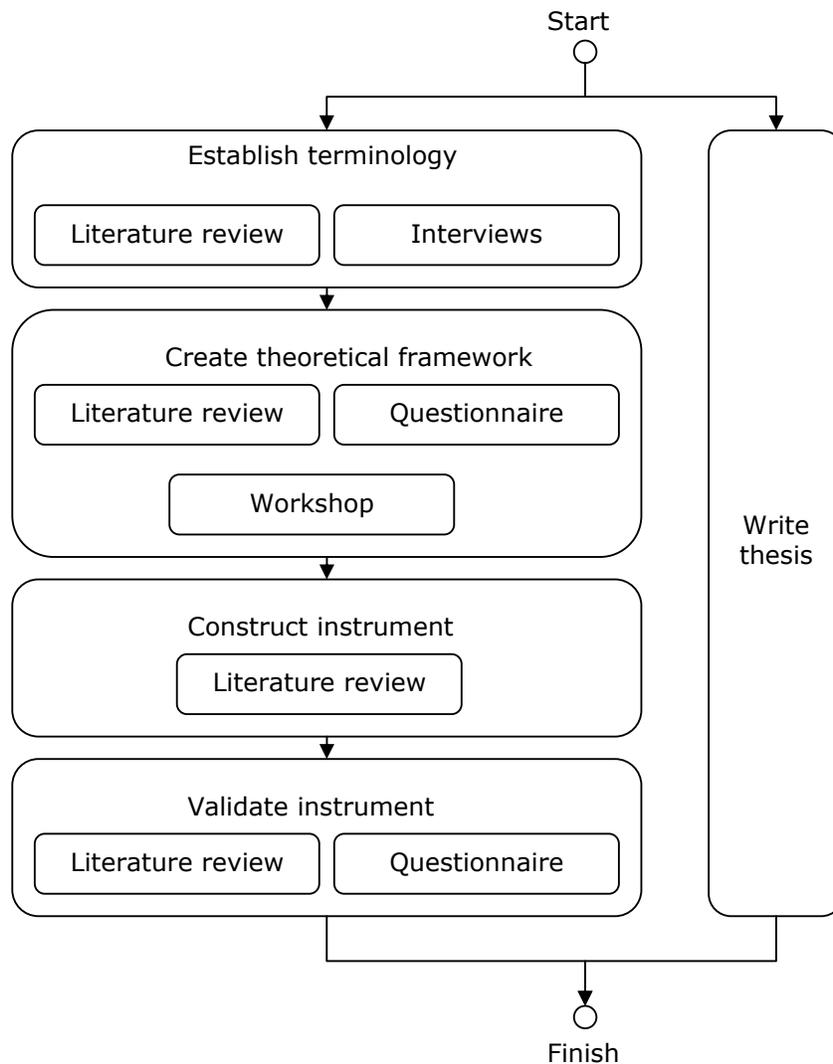


Figure 1: Thesis work plan and research methods

#### *Literature reviews:*

In all phases of this thesis project, literature reviews have played an important role. The project initially started with literature review into the topics related to the field of enterprise architecture. Based on this review the goal of the research was formulated. After the research goal was made clear, the literature review was resumed to define the relevant concepts such as enterprise architecture, products and processes and quality. A second literature review was aimed at creating a theoretical framework, consisting of a quality model and process viewpoints. The construction phase was also supported by several literature sources. These sources primarily addressed examples of other studies into instrument development and the construction of instrument scales. The last literature review that was conducted examined approaches to instrument validation.

#### *Expert interviews:*

During the first phase of this thesis project expert interviews were conducted as means to gain more insight into the field of enterprise architecture. A number of experienced architects within Sogeti were asked about their experiences and opinions with respect to architecture and quality [40].

#### *Workshop:*

During the construction of the theoretical framework an interactive workshop was organized to gain more insight into issues relevant to the quality of the enterprise architecture development process [39]. The ten attending professionals were all experienced in the field of enterprise architecture.

#### *Questionnaires:*

During this thesis project, two questionnaires were created. In combination with the workshop describe above, the first questionnaire was created to gain more insight into issues relevant to the quality of the enterprise architecture development process [41]. The seven respondents were all experienced in the field of enterprise architecture. The second questionnaire was used for validating the quality instrument that has resulted from this thesis project [42]. Results from twenty-six respondents from different companies throughout the Netherlands were used in analyzing the reliability and validity of this instrument.

### **1.3 Previous research**

The literature study uncovered little scientific research into the subject of enterprise architecture. As mentioned earlier most of the enterprise architecture approaches available are bundled best-practice experiences of experts in the field. The literature review did uncover a number of instruments for determining the maturity of enterprise architecture within organizations [5, 33, 45]. These instruments were developed as part of existing approaches to enterprise architecture and are also the result of best-practice experiences. However on the combination of quality and enterprise architecture but one source of literature was found written by Bernus [6]. This short paper addresses the relation between enterprise architecture models and ISO9000:2000, focusing on products instead of processes. Because quality and quality management is far more developed in other fields of expertise such as software engineering [25], health care [24] and business process management [19], the knowledge from these disciplines has been an important source of information for this thesis project.

#### **1.4 Scientific & social relevance**

This research tries to answer the main research question that asks how an instrument for measuring the quality of the enterprise architecture process can be created. Such a quality instrument would have to be scientifically validated to establish its reliability and validity. As such it would provide a valuable contribution to both the field of enterprise architecture and the scientific world. The field of enterprise architecture would benefit by having an instrument available to measure the quality of the architecture process in an objective and reliable manner. As such it could be applied in audits, as an independent measure of process quality or as a reliable tool for self-assessment [16].

Science would benefit from this research in two ways. Firstly by providing it with a reliable instrument that could be used as a tool in benchmark studies of the quality of the enterprise architecture process. The results of such benchmark studies could help to gain more insight into enterprise architecture and as such contribute to maturing the field. Secondly science could benefit from the work plan followed in the development of the instrument and the theoretical framework this instrument would be founded on. A structured approach towards developing a quality instrument for a process could be repeated for other processes in the field of enterprise architecture or even other fields of expertise. And publication of this instrument and their measurement analysis would allow other researchers to use the same instrument [16].

#### **1.5 Results**

This thesis project has resulted in the following deliverables:

- A list of definitions of the concepts used in the field of enterprise architecture.
- A theoretical framework for creating an instrument that measures the quality of the enterprise architecture development process. This framework consists of a quality model and a set of viewpoint representations relevant to the architecture development process.
- A reliable and valid quality instrument for measuring the quality of the enterprise architecture development process.
- A structured approach towards developing a quality instrument that can be used for developing instruments for other processes.

#### **1.6 Thesis outline**

In part I of this thesis document the theoretical framework that has formed the foundation for the quality instrument for the architecture development process. The different components of this framework are addressed one by one.

Part II of this thesis describes the steps that were undertaken in the construction and validation of the quality instrument. This part also addresses issues that arise in applying this quality instrument.

The conclusions and recommendations for future research are discussed in the third and last part of this thesis, part III.

---

## Part I: Theoretical framework

The first step in developing a quality instrument for the enterprise architecture process consisted of the creation of a theoretical framework. Part I of this thesis document describes the development of this framework. The theoretical framework consists of formal representations of the viewpoints relevant to the enterprise architecture development process. These formal representations will be defined in chapter 2 together with the other concepts used in this thesis. After this, chapter 3 describes the scope of this thesis project. Chapter 4 describes the development of a quality model for the processes in the scope of this thesis project. This model consists of a collection of quality attributes that address process characteristics. In chapter 5 the viewpoints relevant to the processes within the scope are defined. The last chapter of part I describes the theoretical framework by summarizing the viewpoint representations.

---



## 2 TERMINOLOGY

### 2.1 Introduction

The concepts defined in this chapter are the results of an extensive literature review. Figure 2 provides a structured overview of the concepts that will be discussed and their interrelationships. The arrows provide the direction of these relationships. The diamonds indicate a sub division of a concept. For example, the enterprise architecture is divided up into four sub processes: development, maintenance, deployment and governance. In the following sections each of the concepts depicted will be explained and defined.

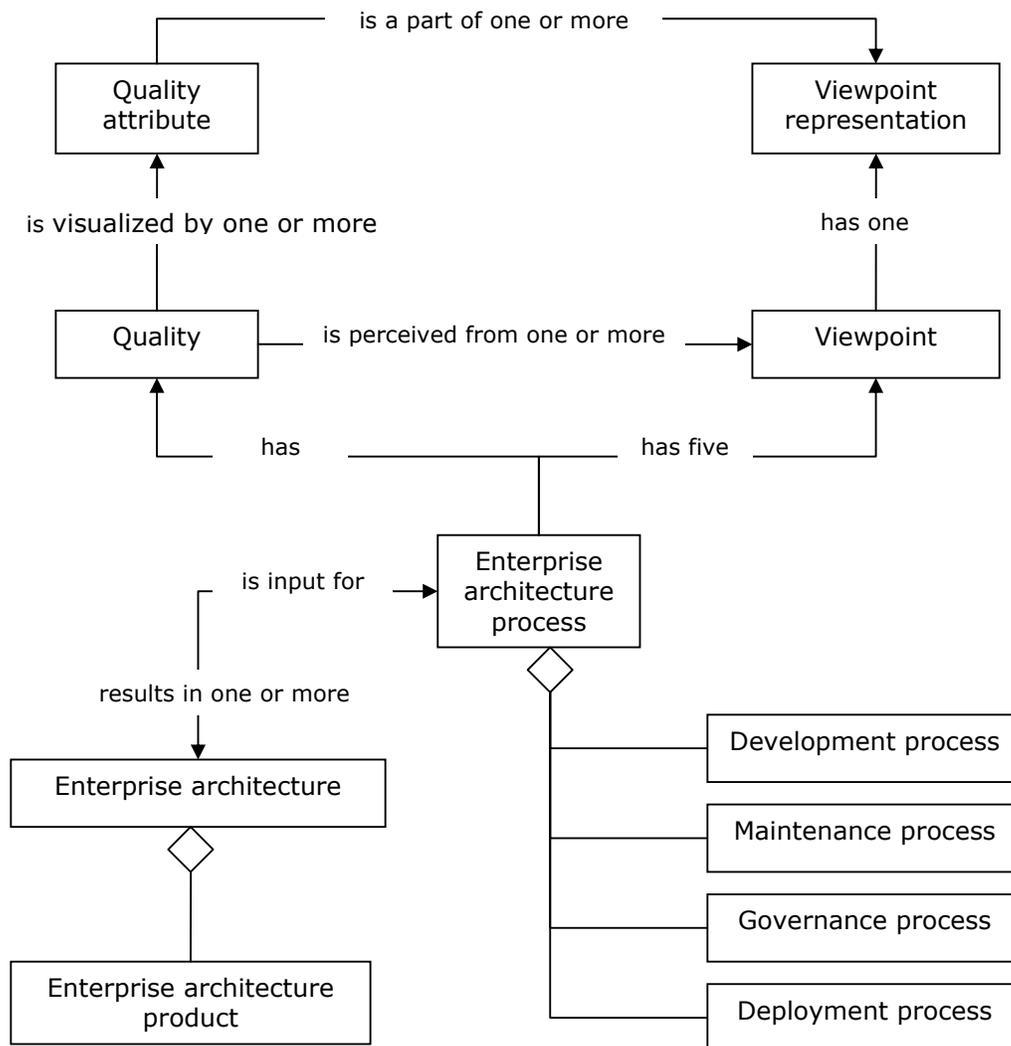


Figure 2: A graphical representation of the concepts used in this thesis.

## 2.2 Enterprise architecture

Until this moment there is no consensus on the definition of enterprise architecture. Many practitioners of enterprise architecture have their own working definitions for this concept. A number of the available definitions are presented below.

Rijsenbrij [31] recently published the definition: *'Digital architecture is a coherent and consistent collection of principles, divided into concerns, rules, standards and guidelines that describe how an enterprise, the information services, the applications and the infrastructure were shaped and how they react during its use.'*

DYA<sup>®</sup> [45] states: *'Enterprise architecture is a consistent whole of principles and models that direct the design and realization of the processes, organizational structure, information services and technical infrastructure of an organization.'*

The Meta Group [9] defines enterprise architecture as: *'A top-down, business-strategy-driven process that coordinates the parallel, internally consistent development of enterprise business, information and technology architectures, as well as the enterprise application portfolio.'*

Hoogervorst [20] sees architecture as: *'a consistent set of design principles and standards that guide design.'*

IEEE standard 1471-2000 [21] states that enterprise software architecture is: *'The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.'*

The definitions mentioned above show similarities, but also some noticeable differences. Part of these differences can be explained by the division of enterprise architecture in prescriptive and descriptive approaches as identified by Rijsenbrij [31] and Hoogervorst [20]. Descriptive architecture focuses on describing the factual solution, whereas the prescriptive view defines the borders within which the solutions need to be realized. Looking at the definitions above, it can be seen that the definitions of Rijsenbrij, Hoogervorst, DYA<sup>®</sup> and the Meta Group adopt the prescriptive view, whereas IEEE works from a more descriptive approach. The prescriptive definitions mentioned above share the acknowledgement of different domains to which architecture is applicable. Rijsenbrij identifies the domains: enterprise, information services, applications and infrastructure. DYA<sup>®</sup> adopts a similar division by identifying the domains: processes, organizational structure, information services and technical infrastructure. The Meta Group identifies the domains: enterprise business, information and technology services and applications. In his paper *'Enabling integration, agility and change'* Hoogervorst [20] elaborates his definition of enterprise architecture by identifying four domains: business, organizational, information and technology. The descriptive definition however does not identify different domains, but merely speaks of *'a system'*.

A remarkable difference between the definition of the Meta Group and the other definitions is the perception of enterprise architecture as a process instead of as a product. In this the Meta Group's perception of enterprise architecture differs considerably from the others. Of the remaining definitions one last difference can be noticed in the nature of enterprise architecture. Rijsenbrij and Hoogervorst define architecture as a set of principles, whereas both IEEE and DYA<sup>®</sup> also include models in their definition of enterprise architecture.

The differences between the existing definitions of enterprise architecture make adopting one of the mentioned definitions difficult. Definition 1 was formulated as the working definition for enterprise architecture for this thesis. This definition is based on the DYA<sup>®</sup> definition. Parallel to this thesis project, Veltman [44] researched the quality of

enterprise architecture product. To maintain a clear separation between the two subjects, this definition for enterprise architecture was altered slightly by replacing the 'consistent whole of principles and models' by 'a set of products', resulting in definition 1.

Definition 1: Enterprise architecture

*Enterprise architecture is a set of products that direct the design and realization of the processes, organizational structure, information services and technical infrastructure of an organization.*

### 2.3 Enterprise architecture process

With a working definition for enterprise architecture in place, the next concept that needed to be defined was the subject of this research, the enterprise architecture process. The literature review uncovered similar problems as were discussed in the previous section. A number of approaches to enterprise architecture from organizations such as the IFEAD [33], The Meta Group [9], The Open Group [36] and Sogeti [5, 45] were uncovered, but a field standard could not be identified. To prevent this thesis project from becoming biased towards one of the available approaches, the choice was made not to adopt one of enterprise architecture process definitions belonging to one of the existing approaches. Next to the problem of this project becoming biased by adopting one of the existing process definitions, there was a second reason for an alternative solution. Choosing one of the existing process definitions would also limit the applicability of the resulting quality instrument. The instrument would only be suited to measure the quality of those architecture processes that had adopted the selected approach. Thus rendering the instrument useless for measuring the quality of other architecture processes.

So instead of adopting one of the existing architecture processes a comparative study of existing architecture approaches was performed. A summary of the examined approaches can be found in Table 1.

Approach:	Created by:
TOGAF's Architecture Development Method (ADM)	The Open Group
DYnamic Architecture (DYA)	Sogeti Netherlands B.V.
Enterprise Architecture Approach (E2A)	IFEAD
Enterprise Architecture Process Model	The Meta Group
Federal Enterprise Architecture (FEA)	U.S. Chief Information Officer Council [11]
For additional information about enterprise architecture:	
The Zachman Framework	John Zachman [47]
IEEE 1471-2000	IEEE [21]

*Table 1: Examined enterprise architecture approaches*

Originally the goal of this comparative study was to identify the sub processes, activities and products of the respective approaches in order to construct a generic but detailed enterprise architecture process. However, the diversity of the available approaches caused problems in defining such a process. Several attempts in defining a generic process resulted in a large collection of sub processes, activities and products with much overlap, but very little consistency among them. Addressing all these issues would have resulted in a solution that was unworkable because of its size and rigid nature. Another reason for discontinuing along this path was the ongoing discussion about the definition of enterprise architecture. Even if this approach to defining the enterprise architecture process would result in a workable solution, a process description of this nature would most likely be the subject of elaborate discussion for some time.

These were the reasons for choosing an alternative solution in the form of a more high-level definition of the enterprise architecture process. After having identified the different building blocks of all the selected architecture approaches, these blocks were grouped into four sub processes of the enterprise architecture process: development, maintenance, deployment and governance. And although this solution provides a high-level working definition, this working method has provided substantial insight into the details of the identified sub processes. These insights would prove to be of value at a later point in time during this project. Based on this grouping the following working definition for the process of enterprise architecture was constructed.

Definition 2: The enterprise architecture process  
*The enterprise architecture process consists of the activities involved in the development, maintenance, deployment and governance of enterprise architecture.*

Rijssenbrij [31] divides the enterprise architecture process in two distinct parts that correspond to the development and deployment sub processes. Definition 2 adds two sub processes: maintenance and governance. Figure 3 shows a graphical representation of definition 2 and provides some insight into their inter-relationships. The following sections address the four sub processes that were identified for the enterprise architecture process in more detail.

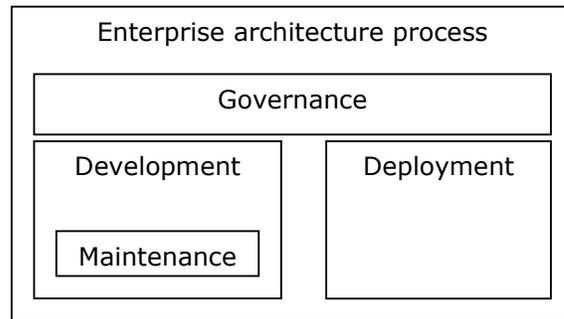


Figure 3: The sub processes of the enterprise architecture process

Figure 3 provides an understandable graphical representation of the enterprise architecture process. It will be used in a later section to clarify the scope of the theoretical framework and the resulting quality instrument.

### 2.3.1 Development process

The development process encompasses all sub processes, activities and products that relate to the creation of enterprise architecture. Requirements, constraints, architectural principles, etc. form the input for this process that results in one or more enterprise architecture products. These products in turn are used in the deployment process as part of, for example projects and organizational change and improvement.

### 2.3.2 Maintenance process

The maintenance process is the second sub process of the enterprise architecture process and includes all sub processes, activities and products that relate to maintaining an up-to-date and consistent repository of EA products. The products resulting from the development process are the input for this sub process. The products stored in the repository are in turn provided to the development process as input. In this research the maintenance is seen as part of the development process, because of the close cooperation and dependency between these two processes.

### 2.3.3 Governance process

The next sub process that was identified is the governance process. Governance includes all sub processes, activities and products that relate to monitoring and organizing the enterprise architecture process and all its sub processes, activities and products. The governance process focuses its attention on the other sub processes of the enterprise architecture process. It is a process of constant monitoring and (re)organizing of the development, maintenance and deployment processes. In this thesis the focus for this process will be on the effort related to governing the development and maintenance processes.

### 2.3.4 Deployment process

The deployment process is the last sub process that was identified for the enterprise architecture process. This process encompasses all sub processes, activities and products that relate to the deployment of the created and/or stored enterprise architecture products. This process was not included in the scope of this thesis and will for that reason not be discussed in depth in this thesis.

## 2.4 Enterprise architecture products

The sub processes and activities of the enterprise architecture process result in and make use of tangible products, the enterprise architecture products. Veltman [44] has defined these products as follows in her thesis.

Definition 3: Enterprise architecture products [44]  
*Enterprise architecture products constitute a set of tangible enterprise domain reflections in either language or visualizations.*

## 2.5 Quality

A literature study into the subject of quality revealed a number of different definitions for the concept quality. The ISO 8402 vocabulary states that quality is: *'the totality of feature and characteristics of an entity that bear on its ability to satisfy stated and implied needs'* [7]. The US ISO 9000:2000 standard for quality management and quality assurance defines quality as: *'the degree to which a set of inherent (existing) characteristics fulfils requirements'* [22, 6]. Beamon et al. [4] researched the subject of quality and uncovered a number of definitions that use terms like 'fitness for use' and 'conformance to requirements'. All of the uncovered quality definitions include one or more of the three components of the ISO 9000 definition identifies: features and characteristics, requirements and fit. Beamon et al. [4] supplement these ideas by saying that the quality requirements are defined by the expectations of the customer. This introduction of the customer and the importance of its expectations on the perception of quality would come to play a big role in the rest of this thesis project. However, it is the opinion of this researcher that the customer is not the only stakeholder for which expectations translate into requirements. This has resulted in definition 4 as the working definition for quality. This definition combines the issue addressed above into one working definition.

Definition 4: Quality

*Quality is the totality of features or characteristics of an entity that bear on its ability to fulfill the requirements of all stakeholders.*

### 2.5.1 Quality attributes

Definition 4 speaks of features and characteristics. In the field of software engineering such features and characteristics are often referred to as quality attributes. Wijnstra [46] defines quality attributes in the context of software engineering as: *'observable properties'*. Based on this definition the working definition for quality attributes was formulated as:

Definition 5: Quality attribute

*A quality attribute is an observable property of a process or activity.*

### 2.5.2 Quality of the enterprise architecture process

Definition 6 provides the working definition for the quality of the enterprise architecture process. It was constructed by applying definition 4 to the enterprise architecture process.

Definition 6: Quality of the enterprise architecture process

*The totality of features and characteristics of the enterprise architecture process that bear on its ability to satisfy the stated or implied requirements of its stakeholders for enterprise architecture.*

### 2.5.3 Quality of the enterprise architecture products

For the quality of enterprise architecture product, the definition of Veltman [44] was adopted.

Definition 7: Quality of enterprise architecture products  
*The quality of enterprise architecture products is represented by the totality of attributes that bear on the EA product's ability to aid integration and agility in a consistent and coherent way.*

## 2.6 Viewpoints

A process can be perceived differently by different stakeholders. Mullery [26] was one of the first to acknowledge the importance of these different perspectives. He introduced the concept of viewpoints to handle different perspectives in the field of requirements specification. Several other authors also acknowledge the importance of different perspectives. Finkelstein et al. [14] employs viewpoints as part of a framework for supporting multi-perspective system development. This same Finkelstein [15] wrote a FAQ on the subject of viewpoints in which he addresses a number of the most frequently asked questions on this subject. Sommerville et al. [34] and IEEE std. 1471 [21] are two more examples of authors that acknowledge the importance of identifying different perspectives.

In this thesis project the concept of viewpoints was introduced to represent the different stakeholders and their requirements in the context of quality measurement. A group of stakeholders with the same concerns or requirements is represented by one viewpoint. This has resulted in definition 8 for a viewpoint.

Definition 8: Viewpoint  
*A viewpoint represents the perception of a process by one or more stakeholders.*

Definition 8 speaks of *the perception*, but how can perception be represented? To answer this question the examples of Sommerville et al. and IEEE std. 1471 were followed. Sommerville et al. provides a semi-formal, semi-flexible vision on viewpoints. They introduce viewpoints as a means to identify and manage inconsistencies in software development processes. Their viewpoints are represented by a tuple of five components: a name, a set of concerns, information sources, a focus and a process description. The first component *name* provides a unique identifier for the viewpoint. The *concerns* of a viewpoint address the business goals and constraints that drive the software process. A viewpoint's *focus* provides its respective perspective on the process. The *information sources* are defined to make it easier to gather the necessary information concerning the process. And last the *process description* provides a hierarchical description of the process from the perspective of the viewpoint.

IEEE std. 1471 explains a similar approach to representing viewpoints. This standard provides a recommended practice for architectural descriptions of software-intensive systems in the field of software system development and evolution. The concepts of views and viewpoints are introduced in this standard as a means for addressing the concerns of different stakeholders in a structured manner. A view represents: '*the expression of a system's architecture with respect to a particular viewpoint*'. And this viewpoint provides the formal conventions by which a view is created, depicted and analyzed.

Each viewpoint is represented by four components: a *name*, its *stakeholders*, their associated *concerns* and *language and modeling techniques*. Again the *name* is used as a unique identifier. Every viewpoint has one or more *stakeholders* that share the same perspective associated. These stakeholders share the *concerns* associated with the viewpoint. The last components of a viewpoint are the *language and modeling techniques*. These consist of guidelines to be used in constructing a view based on a viewpoint.

The viewpoint representations by Sommerville et al. and IEEE were used in determining a suitable representation for the viewpoints of the enterprise architecture process. This representation will be explained in the next section.

### 2.6.1 Viewpoint representation

The previous section provided the definition of a viewpoint, definition 8, and the two existing representations of a viewpoint by Sommerville et al. [34] and IEEE [21]. This section describes the viewpoint representation for the enterprise architecture process as it was constructed from these to examples to benefit this thesis project.

Neither of the two described viewpoint representations was found to be one hundred percent applicable to represent the different perspectives on the architecture process. The main problem with the IEEE representation is the nature of the subject it addresses. An architectural description of a software-intensive system is a product, whereas viewpoints in this thesis project are used for different perspectives on a process, the architecture process. In the IEEE std.1471 viewpoints form a formal template for describing a view by using the prescribed language and modeling techniques. Software systems are highly suited for formal description, but the enterprise architecture process requires more flexibility in its representation, since nothing in this process standard.

The representation of Sommerville et al. provides this flexible solution, but also has a shortcoming. Similar to the central subject in this thesis, Sommerville et al. define viewpoints for a process, a software process. But the problem lies in the way viewpoint concerns are addressed. Sommerville et al. identify the concerns of a viewpoint as business goals and needs and relate them to process changes and improvements. First, although improvements and/or changes might result from using the quality instrument, it was not the main focus for using viewpoints. And second, the concerns related to the different viewpoints of the enterprise architecture process might address other issues than merely business goals or needs.

By combining both of the described representations a viewpoint representation was developed that is both flexible and informal enough to be applied to the enterprise architecture process. The representation of Sommerville et al. with its five components was taken as the foundation: *name*, *focus*, *source*, *process description* and *concerns*. The *name* component was maintained as a unique identifier. The *focus* was also maintained to provide the perspective of the respective viewpoint. The *process description* and the *sources* were deleted from the representation. Sommerville et al. included a process description in their viewpoint representation. As was addressed in section 2.3, it is very difficult to define a generic and detailed enterprise architecture process. It would be just as difficult to describe the architecture process for the different viewpoints in a generic way or even to provide some sort of generic process mapping, as was proposed by Anjard [2]. Because it is the goal of this thesis to develop an instrument for measuring quality that is applicable to all architecture development processes, the process description will be omitted from the viewpoint definition. The *information sources* were removed because they did not provide a valuable addition to the concept, since architects involved in the development process most likely know where to find the necessary information. A second reason for removing the *sources* is

that the terminology in the field of enterprise architecture is so ambiguous that a list of possible information sources would only result in confusion and discussion. The *concerns* form the last component of the viewpoint representation. As stated earlier the *concerns* of Sommerville et al. do not suffice for defining the concerns of the architecture process viewpoints. And this is where IEEE std 1471 provides the solution. IEEE defines the concerns of a viewpoint in the form of general concerns associated to the stakeholders. For that reason, IEEE's *concerns* were adopted to represent all issues of concern for the stakeholders of the viewpoint, not just business goals or needs. One last addition to the representation of a viewpoint was the *stakeholder* component of IEEE std. 1471. Explicitly listing the stakeholders represented by a viewpoint should improve understanding of the perspective it represents.

Summarizing, the representation of a viewpoint that was defined in definition 8, consists of a tuple that contains four components: *name, stakeholders, focus and concerns*.

*Name:*

A viewpoint of the enterprise architecture process is uniquely identified by its name.

*Stakeholders:*

This component of a viewpoint representation identifies the one or more stakeholders that share the same perspective on the enterprise architecture process

*Focus:*

The focus of a viewpoint describes the perspective of that viewpoint on the process. It provides an indication of the issues that are important for the respective viewpoint.

*Concerns:*

The concerns of a viewpoint represent all issues that are relevant for the viewpoint perspective. In the context of this thesis project concerns address the issues relevant to the respective viewpoint in the perception of enterprise architecture process quality from that viewpoint. These relevant issues will be represented by a set of quality attributes.

### 3 SCOPE

---

With the key concepts defined, the scope of the theoretical framework needs to be addressed. Originally the goal of the thesis project was to develop a quality instrument for the enterprise architecture process as it was defined in definition 2. However, the enterprise architecture process can be divided into two parts, each with very distinct properties. Rijsenbrij [31] makes this distinction by dividing the enterprise architecture process into: *'the architecturing'* and *'the transformation to come to the artifacts that meet the proposed architecture'*. These two parts correspond to the development and deployment sub processes that have been identified during this thesis project. The distinct difference between processes that create a product and the processes that deploy that same product was the reason for narrowing the scope of the thesis project to one of the two parts.

The choice was made to focus on the deployment process and its sub processes and activities. Figure 4 provides a graphical representation of the scope of this thesis project. The red box shows the processes that were included in the scope. The focus of this thesis project is on the deployment process. And since maintenance was defined as a sub process of the deployment process, this process also falls within the scope. Next to these two processes, the scope also encompasses part of the governance process. The governance process consists of all sub processes, activities and products related to monitoring and organizing the enterprise architecture process and all its sub processes, activities and products (section 2.3.3). Since part of these activities relate directly to either the deployment or the maintenance process, part of governance has also been included in the scope of the thesis project. To benefit the readability of this document, from this point on the sub processes and activities within the scope of this project will be addressed as the development process.

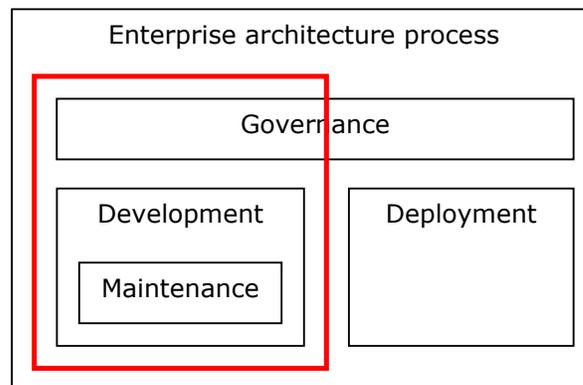


Figure 4: The focus of the theoretical framework



## 4 QUALITY MODEL

---

### 4.1 Introduction

In the field of software engineering, quality models are used to define the quality of software systems and software engineering processes. Ortega et al. [29] discuss the evolution of quality models over the years by addressing amongst others McCall, FURPS and ISO 9126. They see a quality model as a set of features and characteristics (attributes) with corresponding metrics, divided over one or more dimensions, to define the quality of the process or product under investigation. This chapter describes the creation of a quality model for defining the quality of the enterprise architecture development process. This quality model will be used when the concerns of the viewpoints associated with the architecture development process viewpoints are addressed. The three literature sources that have formed the basis for this model are described one by one together with their individual contributions to the resulting quality model in figure 4. The metrics corresponding to the quality attributes will not be addressed in this chapter, but will be explained in chapter 7.

### 4.2 Quality model I

Satpathy et al. [32] adopted the use of quality attributes to determine the quality of the development and maintenance processes of software. Based on ISO/IEC 9126 & 12207, ISO 9001.9000-3, CMM and the FURPS+ model, they constructed a quality model that distinguishes eight quality dimensions. The thirty-three quality attributes that make up these eight dimensions were defined for both the software products as the software development and maintenance processes. This is where the paper by Satpathy et al. differs from other papers. Most literature on quality models [8, 10, 17] focuses on the quality of products, but Satpathy et al. also address the quality of processes in their attribute definitions. For that reason the quality model by Satpathy et al. was chosen as the foundation for the quality model for the architecture development process.

#### 4.2.1 Quality dimensions

Quality models are often given a hierarchical structure by grouping related quality attributes. These groupings are referred to as quality dimensions. This type of structuring was also adopted for the quality model of the enterprise architecture development process. Inspired by the eight dimensions of Satpathy et al. the following seven dimensions were selected to make up this model.

##### *Functionality:*

The quality attributes in this dimension address the existence of functions that are available in the process. In the context of the enterprise architecture development process these functions satisfy stakeholder requirements.

##### *Usability:*

The attributes in this dimension address the efforts required for understanding, learning to use and operating the architecture development process.

##### *Efficiency:*

The attributes in this dimension address the performance of the development process with respect to the use of time and resources.

##### *Visibility & Control:*

The quality attributes in this dimension address the possibilities to monitor and control the development process.

*Reliability:*

The attributes in this dimension address the frequency of failure and the flexibility of the enterprise architecture development process in addressing and overcoming these failures.

*Scalability:*

The attributes in this dimension address the ability of the development process to maintain functionality and efficiency when handling problems of larger dimensions.

*Maintainability:*

The attributes belonging to this dimension address the effort required for realizing modifications to the architecture development process and the flexibility of the process in handling these modifications.

From the eight dimensions identified by Satpathy et al., the dimension *Safety* was not included. The single attribute that makes up this dimension was found to be too product-oriented to be applicable to the architecture development process.

4.2.2 *Quality attributes*

The thirty-three quality attributes corresponding to the original eight dimensions of Satpathy et al. [32] were reduced to twenty-nine attributes that might prove to be relevant in determining the quality of the architecture development process. These twenty-nine remaining attributes were not directly applicable to the enterprise architecture development process, but needed to be rewritten to conform to the terminology of the field of enterprise architecture and specifically the architecture development process. Divided over the seven remaining dimensions the following twenty-nine quality attributes from the model by Satpathy et al. were included in the quality model for the development process.

*The dimension Usability:*

All three quality attributes that make up the *Usability* dimension of Satpathy et al. were included in the quality model. The attributes *understandability*, *learnability* and *operability* are described below. The literature sources that identify these quality attributes are listed in the last column of the table. These attributes address three levels of process usability: the effort required to understand, the effort required for learning to use and the effort required for executing the development process.

Quality attribute	Definition	Source
Understandability	Indicators that bear on the effort required from an architect to understands the logical concepts of the development process.	[8], [10], [32], [43]
Learnability	Indicators that bear on the effort required from an architect to learn to use the process.	[8], [10], [32], [43]
Operability	Indicators that bear on the effort required from an architect to execute the development process with a level of confidence.	[8], [10], [32], [43]

Table 2: *Quality attributes added to the dimension Usability from quality model I*

*The dimension Functionality:*

The attributes *compliance, completeness, consistency, generality, suitability and inter-operability* were maintained in the dimension *Functionality*. The definitions of these attributes can be found in the table below. The definitions of *generality* and *inter-operability* might need some clarification. First *generality*, this attribute addresses the process' ability to anticipate requirements that were not formulated by the stakeholders at the beginning of the process. An example of such a requirement could be extra flexibility to realize connection with new technologies. The attribute *inter-operability* addresses the contribution the development process makes to the resulting product's ability to function in for example the deployment phase. *Security* was the only quality attribute from Satpathy et al. that was not incorporated. This attribute addresses security against hostile environments. In the context of the enterprise architecture development process, this is an irrelevant attribute.

<b>Quality attribute</b>	<b>Definition</b>	<b>Source</b>
Compliance	Indicators that bear on the degree to which the development process conforms to its own process model.	[8], [10], [32], [43]
Completeness	Indicators that bear on the degree to which the development process transforms all of the input requirements into the resulting enterprise architecture product(s).	[10], [32]
Consistency	Indicators that bear on the degree to which the development process does not introduce contradictions in the resulting enterprise architecture product(s).	[10], [17], [32]
Generality	Indicators that bear on the ability of the development process to address conditions which are not covered by the input requirements but are relevant to the context in which the resulting enterprise architecture product(s) will be used.	[17], [32]
Suitability	Indicators that bear on the degree to which the development process matches the resulting enterprise architecture product(s) adequately to the input requirements.	[8], [10], [19], [32], [43]
Inter-operability	Indicators that bear on the degree to which the development process contributes to the resulting enterprise architecture product's ability to function in its later environments.	[8], [10], [17], [32]

Table 3: Quality attributes added to the dimension *Functionality* from quality model I

*Efficiency:*

The next dimension is *Efficiency*. Satpathy et al. describe six quality attributes for this dimension. Of these the four attributes shown below were included in the quality model. *Time & resource behavior* are typical efficiency issues and do not need further explanation. *Complexity* and *process maturity* are less straight-forward. *Complexity* goes to the process' ability to handle complex situations; it addresses the flexibility of the development process in handling different types of complexity, such as for example: complexity of input requirements and/or the resulting enterprise architecture product(s). *Process maturity* addresses the maturity level of the enterprise architecture process according to a AMM. In software engineering the Capability Maturity Model [17] is a generally accepted instrument for determining the maturity of software processes. As was mentioned in the introduction of this thesis, there exist a number of similar maturity models that address the architecture maturity of organizations [5, 33, 45]. Although none of these maturity models are as generally accepted as the CMM is in the field of software engineering, the attribute *process maturity* was explicitly included in the model. The possibility to relate the architecture maturity to the process quality might prove to be very valuable. The remaining two quality attributes from the model by Satpathy et al. were so closely related to the four attributes described above, that they were omitted from the quality model for the development process.

<b>Quality attribute</b>	<b>Definition</b>	<b>Source</b>
Time behavior	Indicators that bear on the usage of time by the development process to realize the resulting enterprise architecture product(s).	[8], [10], [19], [32], [43]
Resource behavior	Indicators that bear on the usage of resources by the development process to realize the resulting enterprise architecture product(s).	[8], [10], [19], [32], [43]
Complexity	Indicators that bear on the ability of the development process to support various forms of complexity.	[32], [43]
Process maturity	Indicators that bear on the AMM (Architecture Maturity Model) level of the organization.	[8], [32], [43]

Table 4: Quality attributes added to the dimension *Efficiency* from quality model I

*Scalability:*

Satpathy et al. defined but one quality attribute for the dimension *Scalability*, namely the attribute *scalability*. As the definition states *scalability* addresses the ability of the development process to handle problems of larger dimensions. With problems of larger dimensions one might think of projects that are more time consuming, projects with greater complexity or projects with a bigger impact than regular projects.

<b>Quality attribute</b>	<b>Definition</b>	<b>Source</b>
Scalability	Indicators that bear on the degree to which the development process maintains its efficiency in terms of time, cost and resource usage in handling a problem of larger dimensions.	[10], [17], [32]

Table 5: Quality attributes added to the dimension *scalability* from quality model I

*Visibility & Control:*

All attributes from the dimension *Visibility & Control* as defined by Satpathy et al. were included in the quality model. This dimension includes the attributes listed in the table below. These attributes focus on monitoring the process progress and performance. The feedback data is used for direct action or future improvement.

Quality attribute	Definition	Source
Automatic checks & feedback	Indicators that bear on the ability of the development process to provide feedback data and to support corrective actions if necessary.	[32]
Progress monitoring	Indicators that bear on the ability of the development process to facilitate monitoring at any point of time during its execution to show that progress so far has been correct and effective.	[32]
Improvement measures	Indicators that bear on the ability of the development process to support the analysis of the feedback data in combination with the data of previous runs and improve itself, or result in the improvement of a sibling process, continuously.	[32]

Table 6: Quality attributes added to the dimension *Visibility & Control* from quality model I

*Reliability:*

The dimension *Reliability* was also included in the quality model for the architecture development process as a whole. Its three attributes all go to the process' ability to record the frequency of process faults and failures and handle these process faults or failures.

Quality attribute	Definition	Source
Fault tolerance	Indicators that bear on whether the development process can still continue in the presence of faults/failures in the process itself.	[8], [10], [32], [43]
Failure frequency	Indicators that bear on the number of (and the interval) between failures encountered during the execution of the development process. The failures are due to defects in the design / implementation of the process itself.	[32]
Recoverability	Indicators that bear on whether the development process can attain its level of operation after a process failure has been addressed.	[8], [10], [17], [32], [43]

Table 7: Quality attributes added to the dimension *Reliability* from quality model I

*Maintainability:*

The last dimension is *Maintainability*. All nine original quality attributes were maintained for this dimension. The table below provides their definitions and origin. *Maintainability* consists of nine attributes that address different issues of the maintainability of the development process. *Analyzability*, *defect trend*, *testability* and *(in)formal verification* address the observable transparency of the process.

*Modifiability, stability and reusability* address flexibility related to process maintenance issues. And *portability* goes the flexibility of the resulting products.

<b>Quality attribute</b>	<b>Definition</b>	<b>Source</b>
Analyzability	Indicators that bear on the effort required from an architect to analyze the cause of a fault, process failure or unexpected feedback data.	[8], [10], [17], [32], [43]
Modifiability	Indicators that bear on the effort required from an architect to address failures, detection of faults and unexpected data during the process execution; or faults discovered at a later time but linked to the process.	[8], [10], [17], [19], [32], [43]
Stability	Indicators that bear on the degree to which addressing a process fault adversely affect the development process itself. (Also: the frequency of changes done to the process.)	[8], [10], [32], [43]
Testability	Indicators that bear on the degree to which the development process could be validated itself.	[8], [10], [17], [32], [43]
Defect trend	Indicators that bear on the trend of defects that are observed in the development process itself.	[32]
Formal verification	Indicators that bear on the effort with which properties like consistency, correctness and invariant conditions of the enterprise architecture product under transformation could be formally verified during process execution.	[32]
Informal verification	Indicators that bear on the effort with which properties like completeness, generality etc. of the enterprise architecture product under transformation could be informally verified during process execution.	[32]
Reusability	Indicators that bear on the degree to which components of the development process can be reused in a different context.	[8], [17], [32], [43]
Portability	Indicators that bear on the degree to which the development process facilitates the migration of the product to a different environment.	[8], [10], [17], [32]

Table 8: Quality attributes added to the dimension Maintainability from quality model I

This concludes the contribution of Satpathy et al. to the quality model for the enterprise architecture development process. This section has resulted in an initial model consisting of twenty-nine quality attributes divided over seven dimensions.

### 4.3 Quality model II

By broadening the scope of this literature study the second source of inspiration was found in Harrington's book '*Business Process Improvement*' [19]. Harrington describes features and characteristics (attributes) that determine the quality of business processes. He groups these attributes into three dimensions: effectiveness, efficiency and adaptability. The attributes in the dimension *Effectiveness* address the fit between the need of the customer and the products the process delivers. The attributes in this dimension correspond to the *Functionality* dimension described in the previous section. The second dimension Harrington identifies is *Efficiency*. This dimension focuses on the efficiency of the process throughput and the attributes correspond to the attributes of the dimension *Efficiency* from Satpathy et al. The last dimension Harrington identifies is the dimension *Adaptability*. *Adaptability* addresses the flexibility of a business process. The attributes grouped in this dimension correspond to attributes in the dimensions *Maintainability*, *Reliability* and *Scalability* described in the previous section.

#### 4.3.1 Quality attributes

Harrington [19] provided three additional quality attributes to the quality model. It also provided confirmation of the relevance of a number of the quality attributes that were identified by Satpathy et al. The three additional quality attributes were placed in the dimension *Functionality*. The definitions of these attributes can be found in the table below.

Quality attribute	Definition	Source
Timeliness	Indicators that bear on the degree to which the development process realizes the delivery of the resulting enterprise architecture product(s) within a specified range in time.	[19]
Responsiveness	Indicators that bear on the degree to which the development process is designed to answer to service calls from its stakeholders.	[19]
Repeatability	Indicators that bear on the ability of the development process (steps) to be repeated frequently and consistently.	[19]

Table 9: Quality attributes added to the dimension *Functionality* from quality model II

#### 4.4 Quality model III

The third important source of information was the work of Van Zeist et al. [43]. They developed the Quint2 model for measuring software quality. This quality model is based on the ISO 9126 standard that distinguishes 6 quality dimensions, each split up in two or more attributes, to determine the quality of software products. And although the focus of this model is on the quality of software products, it only required small adjustments to make a selection of these attributes applicable for determining the quality of the architecture development process.

##### 4.4.1 Quality attributes

Based on the Quint2 model one more quality attribute was added to the quality model. The attribute *Clarity* was added to the *Usability* dimension, its definition is shown below. Next to this contribution, five out of the six dimensions that make up the Quint2 model, correspond to the dimensions in the quality model by Satpathy et al. and as a result also in the quality model for the enterprise architecture process. The same can be said for many of the quality attributes that have been included in this model. The references to the literature sources can be found in the tables next to the attribute definitions.

<b>Quality attribute</b>	<b>Definition</b>	<b>Source</b>
Clarity	Indicators that bear on the effort required to make an architect aware of the functions the development process can perform.	[8], [10], [43]

Table 10: Quality attributes added to the dimension Usability from quality model I

## 4.5 Summary

Next to the three literature sources described above the work of a number of other authors in the field of software quality was examined [3, 10, 17]. And although no attributes were added based on these sources, they served as an extra confirmation of possible relevance of the selected attributes to the quality of the development process. The results can be found in the last columns of the tables shown in the previous sections. Figure 5 shows a graphical overview of the resulting quality model, its seven dimensions subdivided into thirty-three quality attributes. A complete list of definitions for all quality attributes can be found in Appendix I.

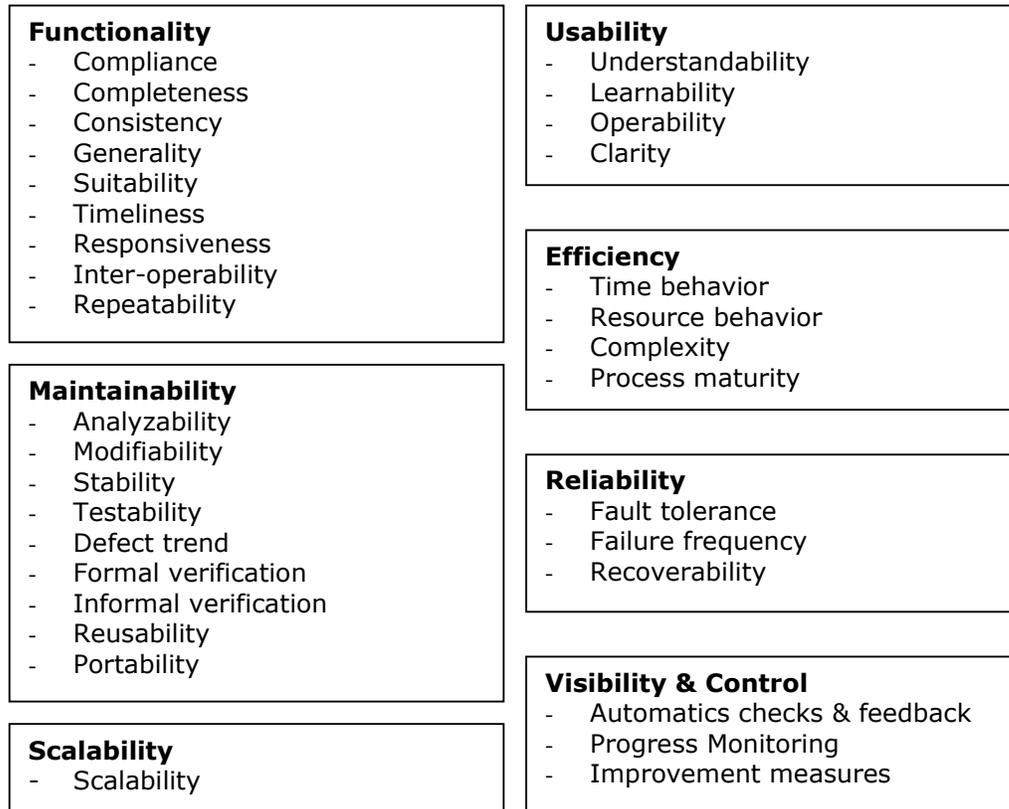


Figure 5: the quality model consisting of 33 attributes divided over 7 dimensions



## 5 VIEWPOINTS

---

### 5.1 Introduction

In this chapter the viewpoints for the enterprise architecture development process, as defined in chapter 3, are identified and represented according to the representation described in section 2.6.1. A viewpoint represents a group of one or more stakeholders that share the same perspective of the development process. The stakeholders joined together in a viewpoint also share their expectations (requirements) of this process. Definition 4 defined quality as: *'the totality of features or characteristics of an entity that bear on its ability to fulfill the requirements of all stakeholders'*. So if the stakeholders in a viewpoint share their requirements for the development process, their perception of process quality will also be the same. It needs to be said that the stakeholders of a viewpoint not exclusively belong to that one viewpoint. Depending on the size of the company, the maturity of their enterprise architecture process and the way this process is organized, one stakeholder might be part of two or more viewpoints. Section 2.3 described the comparative study that formed the foundation for defining the enterprise architecture process. This study also provided valuable insight into the stakeholders of the development process and their corresponding perspectives. This knowledge resulted in the identification of five different viewpoints for the enterprise architecture development process. These five viewpoints will be explained in the following sections. And put together, these viewpoints representation should provide a representative picture of the quality of the architecture development process.

### 5.2 Viewpoints

#### 5.2.1 Customer

The first viewpoint has been named *customer*. The stakeholders united in this viewpoint are future users of the enterprise architecture products that result from the development process. They are responsible for providing (part of) the requirements for these product. This implies that the focus of these stakeholders is primarily on the output of the development process. From their perspective the development process is a black-box that is fed requirements and delivers architecture products.

#### 5.2.2 Executor

The second viewpoint, named *executor*, unites the architects responsible for creating the enterprise architecture products based on the requirements, constraints, etc. while taking into account the architectural principles. The focus of these stakeholders is on the development process itself and the products it delivers.

#### 5.2.3 Maintainer

The viewpoint named *maintainer* is a special type of viewpoint. It was defined to represent one of the sub processes within the scope of this research, the maintenance process. The relatively small set of activities associated with the maintenance process and their strong dependency upon the development process justified looking at maintenance as a viewpoint, instead of as a separate process. This simplified the theoretical framework and as a result the quality instrument. And because quality issues related to the maintenance process can be traced directly back to the development process, this choice did not influence the validity of the quality instrument.

The stakeholders united in this viewpoint are architects responsible for maintaining an up-to-date and consistent repository of architecture products. Their main focus is on

the output of the development process that might need to be stored in the architecture repository. Next to these activities, they might also receive requests to provide executing architects with existing architecture products that are to be used during development.

#### 5.2.4 Governor

The *governor* viewpoint has been defined in a similar way as the *maintainer*. The governance process organizes and monitors the enterprise architecture process and all its sub processes and activities. Within the scope of this thesis project its focus is on the development and maintenance processes. Governance with respect to these processes is realized by implementing monitoring procedures, performance measures, etc. into these sub processes. By defining a *governor* viewpoint to address these issues for both processes, the need for governance to be looked at as a separate process was removed. Again, it resulted in a more simple theoretical framework without damaging consequences for its validity or that of the resulting quality instrument.

The stakeholders united in this viewpoint are responsible for organizing and monitoring the pre-defined enterprise architecture development process. Their main focus is realizing an efficient and effective development process.

#### 5.2.5 Owner

The last viewpoint that was defined for the development process is the process *owner*. Although not directly involved in the development process, the stakeholders united in this viewpoint are responsible for setting the long term (tactical and strategical) goals and needs that enterprise architecture has to facilitate. It is their responsibility to calculate the impact of enterprise architecture. The decisions made by the stakeholders united in this viewpoint have structural consequences for the enterprise architecture process and its sub processes, activities and products.

### 5.3 Concerns

There is one key component missing from the viewpoint description in the previous sections. And that component is the set of concerns associated with every viewpoint. As was mentioned in section 2.6.1, the concerns of a viewpoint with respect to the quality of the development process are represented by a set of quality attributes. These attributes were selected from the quality model that was described in chapter 4. In this selection process experienced architects were consulted about their opinions. These experts were consulted in two steps. The first step, a questionnaire, is described in section 5.3.1. The interactive workshop that formed the second step is described in section 5.3.2.

#### 5.3.1 Questionnaire

The first step in determining which attributes were relevant for the respective viewpoints consisted of a questionnaire, transmitted to experienced architects. The population of respondents consisted of seven practicing architects with a broad experience in the field of enterprise architecture. After a short explanation of the concepts used, they were asked to rate all the attributes in the quality model, described in chapter 4, about their relevance in determining the quality of the development process. This task was performed five times by every respondent, one time for every viewpoint. The relevance of the quality attributes was rated on a five-point scale, one representing *Redundant* and five representing *Very Important*. For every attribute the average score and its standard deviation over all respondents was calculated. Table 11 shows these results. The table shows two columns for every viewpoint. The first column shows the average score over all respondents. The second column provides the standard deviation of the answers.

	<b>Customer</b>		<b>Executor</b>		<b>Maintainer</b>		<b>Governor</b>		<b>Owner</b>	
<b>Functionality:</b>	<i>AV</i>	<i>STD</i>	<i>AV</i>	<i>STD</i>	<i>AV</i>	<i>STD</i>	<i>AV</i>	<i>STD</i>	<i>AV</i>	<i>STD</i>
Compliance	2.6	0.98	3.3	1.11	3.8	0.41	4.0	1.10	3.2	0.75
Completeness	4.1	0.69	3.9	1.07	4.2	0.75	4.5	0.55	3.7	0.82
Consistency	4.1	0.69	4.4	0.53	4.5	0.84	4.3	0.52	3.8	0.75
Generality	3.4	0.79	3.4	0.79	3.5	0.55	3.7	0.52	3.2	1.47
Suitability	3.5	0.55	4.1	0.41	3.6	0.55	3.4	0.55	3.2	1.48
Inter-operability	4.0	0.82	3.6	0.79	4.0	0.89	3.2	0.75	3.5	1.52
Timeliness	4.4	0.79	3.4	0.53	3.2	0.75	4.5	0.55	3.7	0.52
Responsiveness	4.4	0.79	3.6	1.13	3.5	1.22	4.0	0.89	4.3	0.52
Repeatability	3.4	0.53	3.5	1.13	4.3	0.82	4.3	0.82	3.0	1.10
<b>Usability:</b>										
Understandability	3.4	1.13	4.4	0.79	4.0	0.63	3.2	1.17	2.8	1.17
Learnability	3.1	0.90	4.1	0.69	3.8	0.41	3.3	0.82	3.0	1.41
Operability	3.1	0.90	4.1	0.90	3.8	0.75	3.7	0.52	3.2	1.47
Clarity	3.9	0.69	3.9	0.69	3.8	0.75	3.8	0.75	3.3	1.2
<b>Efficiency:</b>										
Time behavior	3.3	1.25	3.6	0.98	3.0	0.89	4.3	0.52	3.3	1.03
Resource behavior	3.0	1.00	3.1	0.69	3.2	1.17	4.2	0.75	3.5	1.05
Complexity	3.0	1.15	3.4	1.13	3.2	1.17	3.7	0.82	3.0	1.41
Process Maturity	2.6	0.79	3.0	0.82	2.8	0.75	3.8	1.17	3.8	1.17
<b>Visibility &amp; Control:</b>										
Automatic checks & feedback	2.6	0.79	2.9	1.35	2.8	1.47	4.0	1.10	3.0	1.26
Progress monitoring	2.9	0.90	3.0	1.29	2.7	1.21	3.8	1.17	3.5	1.05
Improvement measures	2.7	0.90	2.7	0.95	3.0	1.10	3.7	1.03	2.8	1.47
<b>Reliability:</b>										
Failure frequency	3.2	0.75	3.8	0.75	4.0	1.00	4.4	0.55	4.0	1.22
Fault tolerance	2.7	1.03	3.3	1.37	3.4	1.52	3.8	0.84	2.8	0.84
Recoverability	2.8	1.17	3.2	1.17	3.6	1.67	3.8	0.84	3.0	1.58
Safety	2.8	1.17	3.0	1.26	3.2	1.79	2.8	1.30	2.2	0.84
<b>Scalability:</b>										
Scalability	3.1	1.07	3.0	1.15	3.0	1.26	3.7	0.52	3.3	1.51
<b>Maintainability:</b>										
Analyzability	2.3	0.82	2.8	0.75	4.4	0.55	3.8	0.84	3.2	1.30
Modifiability	2.8	0.98	3.3	1.03	4.2	0.84	3.8	0.84	2.8	1.10
Stability	2.7	0.82	3.3	1.37	4.0	1.22	3.4	1.14	3.2	1.48
Testability	3.3	1.25	3.1	0.90	4.2	1.17	3.5	1.05	3.2	1.33
Defect trend	2.3	0.82	2.8	0.90	3.8	1.10	3.8	0.84	3.5	1.30
Formal verification	2.4	0.79	2.7	1.38	3.2	1.47	3.8	0.75	3.2	0.55
Informal verification	2.9	1.07	2.9	1.07	3.3	1.37	3.2	1.17	3.2	1.33
Reusability	2.4	1.13	2.9	1.07	4.0	0.63	3.7	0.82	3.2	1.17
Portability	2.9	0.90	3.0	1.15	3.8	1.47	3.2	0.75	3.3	1.36

Table 11: Results of the first questionnaire

### 5.3.2 Interactive workshop

An interactive workshop made up the second step in identifying the quality attributes relevant to the development process. The questionnaire results shown in the previous section were used as input for this workshop that was attended by ten experienced architects. The primary goal of this interactive session was to finalize the selection of the relevant quality attributes for each of the viewpoints. After an introduction into the research and a discussion about the concepts used, the participants were first asked to identify the quality attributes they found to be of no importance for measuring the quality of the development process. These attributes were to be ignored during the rest of the session. Every attendant was asked to nominate five quality attribute for removal. The results of this first exercise can be found in the first column of table 12, marked 'Votes'. The eight attributes that received three or more votes were discussed to reach general agreement about the removal of those attributes. Of those eight attributes only one was removed without discussion, the attribute *safety*. In combination with its low questionnaire scores, the decision was made to remove the *safety* attribute from the quality model entirely, since it obviously was irrelevant to the quality of the development process. For that reason it is not mentioned in the quality model depicted in figure 3. The remaining seven quality attributes nominated for removal by the participants were: *compliance*, *process maturity*, *failure frequency*, *recoverability*, *modifiability*, *stability* and *formal verification*. On these attributes the participants were less in agreement. The attributes with five or more votes, *compliance*, *process maturity* and *formal verification*, all relate to forms of formal control. And although the workshop participants agreed on their individual irrelevance, they did stipulate that some sort of formal control is necessary. Because they could not agree on which of these attributes to remove, the decision was made against removing them from the session. The participants were also not in agreement on the four remaining attributes nominated for removal. For that reason all were kept during the session. The second workshop activity was aimed at contributing the remaining thirty-three quality attributes to one or more of the viewpoints. To realize this every participant was asked to adopt one of the viewpoints and select the three most relevant attributes for that viewpoint. The results of this part of the workshop can be found in the last five columns of table 12.

### 5.3.3 Results

The results of both the questionnaire and the workshop were combined into one set of relevant quality attributes for every defined viewpoint. The results of the questionnaire were examined and quality attributes with an average score of 3.5 or higher were included in this set. This lower bound was chosen based on the Likert scale that was used in the questionnaire. The average value of 3.5 was interpreted as the turning point between little relevance and some relevance. From the workshop all quality attributes that had received one or more votes in the second part of the workshop were included. The resulting sets of quality attributes consisted of eight to twenty-five attributes for each of the viewpoints. However, this posed a number of problems. Including all of these attributes in the quality instrument would result in an unworkable solution. Eight quality attributes for each of the five viewpoints with an average of three items to quantify the respective attribute would result in an instrument of at least one hundred twenty items. An instrument that size would be far too extensive and time-consuming to be practical. Another problem that would arise with an instrument this size was the time available for this thesis project. Creating one hundred twenty valid items about the quality of the development process would have required far more time than was available.

As a solution to these issues of size and time the choice was made to select the five to seven most important quality attributes for every viewpoint. Chapter 6 goes into detail about the selections of quality attributes for the five viewpoints.

<b>Functionality:</b>	<b>Votes</b>	<b>Customer</b>	<b>Executor</b>	<b>Maintainer</b>	<b>Governor</b>	<b>Owner</b>
Compliance	5	-	-	-	-	-
Completeness	-	-	-	-	2	-
Consistency	-	1	1	-	1	-
Generality	-	1	-	-	-	1
Suitability	-	1	2	-	1	1
Inter-operability	-	-	-	-	-	-
Timeliness	-	1	-	-	1	1
Responsiveness	-	-	-	-	-	-
Repeatability	-	-	-	-	-	-
<b>Usability:</b>						
Understandability	1	-	1	-	-	-
Learnability	1	-	1	-	-	-
Operability	1	-	3	-	-	-
Clarity	-	-	-	-	-	-
<b>Efficiency:</b>						
Time behavior	1	1	-	-	-	-
Resource behavior	1	-	-	-	-	1
Complexity	-	1	-	-	-	-
Process maturity	8	-	-	-	-	-
<b>Visibility &amp; Control:</b>						
Automatic checks & feedback	2	-	-	-	1	-
Progress monitoring	-	1	-	-	-	-
Improvement measures	2	-	-	1	-	-
Failure frequency	3	-	-	-	-	-
Fault tolerance	1	-	-	-	-	-
Recoverability	3	-	-	-	-	-
Safety	5	-	-	-	-	-
<b>Scalability:</b>						
Scalability	-	-	-	-	-	1
<b>Maintainability:</b>						
Analyzability	-	-	-	1	-	-
Modifiability	3	-	-	1	-	-
Stability	3	-	-	-	-	-
Testability	1	-	-	-	-	-
Defect trend	1	-	-	-	-	-
Formal verification	6	-	-	-	-	-
Informal verification	1	1	-	-	-	-
Reusability	-	-	1	-	-	-
Portability	1	-	-	-	-	-

Table 12: Results of the interactive workshop



## 6 FRAMEWORK

### 6.1 Introduction

This chapter summarizes the theoretical framework that has been constructed piece by piece in chapters two to five. This framework consists of five complete viewpoint representations as defined in section 2.6.1. Section 5.3 described the last step in the construction of these representations by identifying their respective sets of concerns. This chapter describes the key points of the resulting five viewpoint representations that form the theoretical framework. Sections 6.2 to 6.6 describe the sets of concerns associated to the five viewpoints by identifying the quality attributes that form these sets. The final section addresses the completeness of the resulting sets of concerns.

### 6.2 Customer

Figure 4 shows a graphical representation of the *customer* viewpoint of the theoretical framework. On the bottom, it shows the seven dimension of the quality model that together make up the quality model. On top of these dimensions the five viewpoints of the development process are depicted. Figure 4 shows the set of most relevant concerns, consisting of the six quality attributes: *timeliness*, *responsiveness*, *completeness*, *suitability*, *consistency* and *inter-operability*. All of these attributes belong to the dimension *Functionality*, which indicates that the stakeholders of this viewpoint are primarily interested in the resulting architecture products and the way the development process realizes these products.

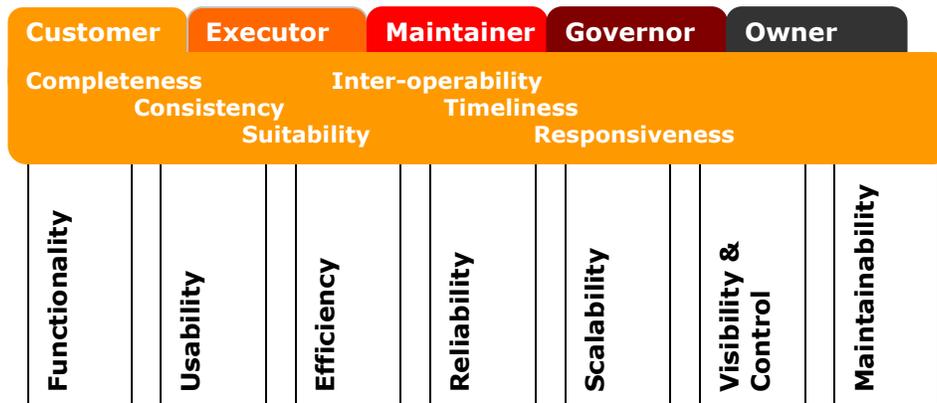


Figure 6: Quality attributes for the Customer viewpoint

### 6.3 Executor

The quality of the development process as it is perceived through the perspective of the process *executor* is best addressed using the quality attributes *consistency*, *completeness*, *suitability*, *understandability*, *learnability* and *operability*. Figure 5 shows the graphical representation. These quality attributes belong partly to the dimension *Functionality* and partly to *Usability*, indicating a mixed interest in results and ease of use. Figure 5 shows a graphical representation of this viewpoint.

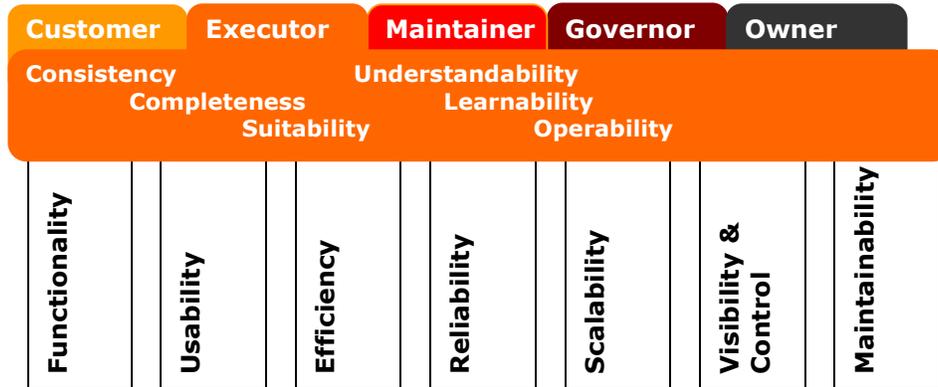


Figure 7: Quality attributes for the Executor viewpoint

### 6.4 Maintainer

Figure 6 depicts the viewpoint representation for the *maintainer*. Its most relevant quality attributes are *consistency*, *completeness*, *repeatability*, *analyzability* and *modifiability*. The attributes *consistency*, *completeness* and *repeatability* belong to the dimension *Functionality*, whereas *analyzability* and *modifiability* belong to the dimension *Maintainability*, again showing a division of priorities over two dimensions.

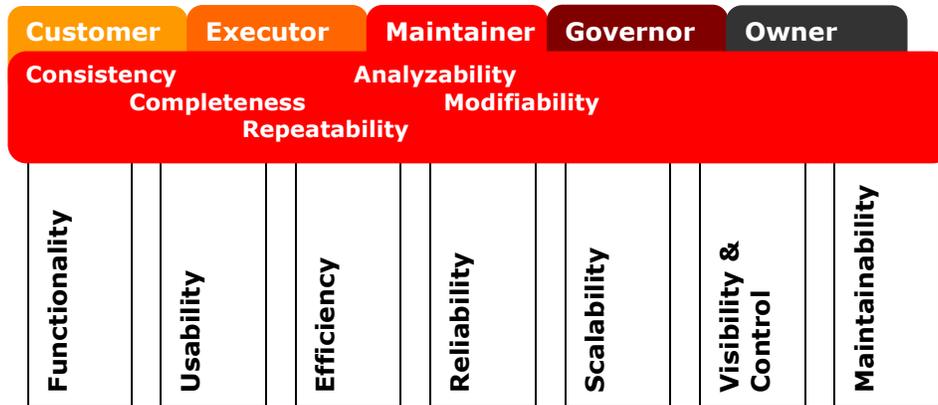


Figure 8: Quality attributes for the Maintainer viewpoint

## 6.5 Governor

The *governor* viewpoint has the most elaborate collection of relevant quality attributes as can be seen in Figure 7. The questionnaire and workshop [37, 39] resulted in a set of 7 most relevant attributes for this viewpoint: *completeness, suitability, consistency, timeliness, repeatability, failure frequency, time and resource behavior*. These attributes are divided over the dimension *Functionality, Reliability* and *Efficiency*.

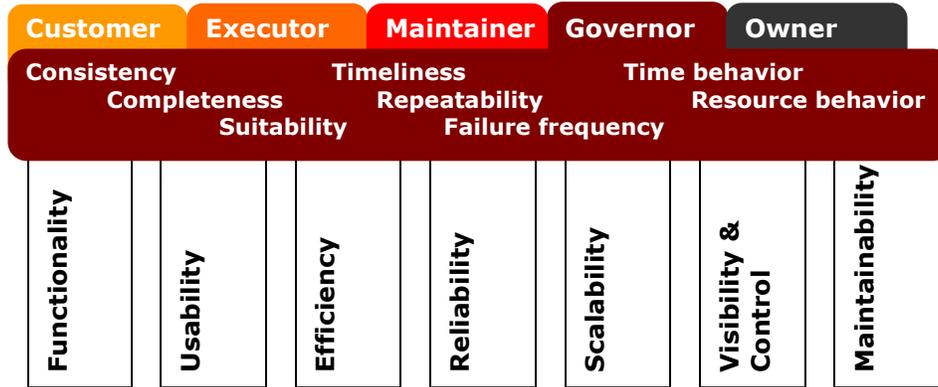


Figure 9: Quality attributes for the Governor viewpoint

## 6.6 Owner

The last viewpoint is that of the process *owner*. And as Figure 8 shows there are five relevant viewpoints for this perspective. It includes the attributes *consistency, completeness, suitability, responsiveness* and *failure frequency*. The attributes *consistency, completeness, suitability* and *responsiveness* belong to the dimension *Functionality*. *Failure frequency* belongs to the dimension *Reliability*.

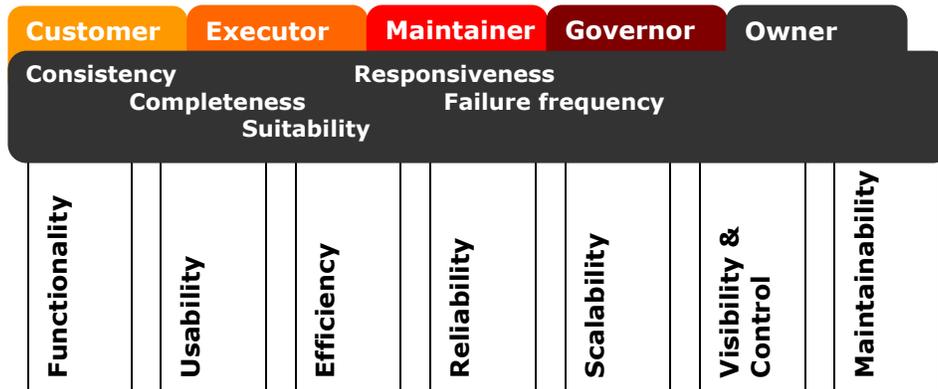


Figure 10: Quality attributes for the Owner viewpoint

## 6.7 Completeness

Due to the choice to address only a selection of the most relevant quality attributes for each of the different viewpoints in the theoretical framework, the resulting quality instrument for the enterprise architecture development process is not complete. Omitting a number of quality attributes that have some relevance to the respective viewpoints has resulted in an incomplete instrument. However in approaching the construction of the quality instrument, this researcher has adopted Pareto's 80/20 rule [38]. This rule has been used in many field of expertise such as for example IT [37], health care [12], finance [38] and even quality management [23]. It states that 20% of the total effort yields 80% percent of the total outcome. In the context of quality measurement for the enterprise architecture development process this rule could be formulated as: the top 20% of relevant quality attribute addresses 80% percent of total process quality. The theoretical framework was constructed by taking the five to seven most relevant attributes for every one of the viewpoints, selected from a total of no more than twenty-five quality attributes with at least some relevance to the quality of the development process. Applying the 80/20 rule to these numbers goes to the conclusion that the instrument measures at least 80% of the total development process quality, since at least the top 20% of all relevant attributes has been selected for inclusion into the instrument. As a result the quality instrument will provide an incomplete, but representative measure for the quality of the enterprise architecture processes it is applied to.

---

## Part II: Quality Instrument

This part of the thesis explains the construction, validation and application of the quality instrument for the enterprise architecture development process. Chapter 7 describes the construction of the quality instrument based the theoretical framework, discussed in the previous chapter. In chapter 8 the statistical validation of the instrument is explained, divided into instrument reliability and instrument validity. The final chapter of part II, chapter 9 addresses the application of the instrument in practice and processing of the results.

---



## 7 CONSTRUCTION

---

### 7.1 Introduction

The viewpoint representations that make up the theoretical framework have been described in chapter 6. Their corresponding concerns, in the form of quality attributes, for measuring the quality of the enterprise architecture development process have been identified. To improve the workability of the resulting instrument, a selection of the most relevant attributes per viewpoint has been made. Based on this theoretical framework, the next step in this thesis project consisted of the construction of the quality instrument. This chapter describes the steps that were undertaken to transform the theoretical framework into a quality instrument. Section 7.2 starts with identifying the sub issues related to the sets of quality attribute associated with every viewpoint. In section 7.3 these sub issues are concerted into items. In the last section of this chapter, section 7.4, the scales of the instrument are explained.

### 7.2 Sub issues

Every viewpoint representation has its own subset of quality attributes. Every one of these quality attributes has its own quality sub issues. These issues address the relevant subjects that should be inspected when measuring the process quality with respect to this attribute. To illustrate these relationships look at the quality attribute *completeness*. The attribute *completeness* is part of the concerns for the viewpoints *customer*, *executor*, *maintainer*, *governor* and *owner*. The total set of sub issues for *completeness* consists of a collection of subjects relevant to these different viewpoints. For example, the *customer* sees realizing maximum completeness as an important sub issue of *completeness*. And although the *governor* is also interested in this sub issue, he also identifies the presence of monitoring procedures as a relevant sub issue. By identifying these sub issues for all the attributes associated with all the viewpoints, a complete collection of subjects is realized that needs to be addressed by the quality instrument.

To aid in identifying the relevant sub issues for each of the attributes the Goal Question Metric approach by Basili et al. [3] was adopted. This approach was designed as a three-leveled model to create measurement instruments for software development. The top (Goal) level of this GQM approach is known as the conceptual level. On the conceptual level a goal for the object of measurement is defined with respect to different models of quality, as perceived from various viewpoints in their own environment. Back to the *completeness* example, one of the goals might be formulated as: measuring the completeness of the development process as it is perceived by the process executor. These goals were formulated for every quality attribute belonging to the concerns of every one of the viewpoints. After completion, the conceptual level of the GQM consisted of twenty-nine goals formulated similarly to the example mentioned above. Every formulated goal represented the combination of a quality attribute and its associated viewpoint and was formulated according to the following format:

Measuring the <<quality attribute>> of the development process as it is perceived by the process <<viewpoint>>.

The second level of the GQM approach formed the main reason for adopting GQM in this part of the thesis project. On this level, the Questions level, a set of questions is formulated to help identify the sub issues relevant in realizing the stated goals. This way GQM provided a structured approach for identifying the sub issues of a quality attribute relevant to measuring the quality of the architecture development process. Questions were formulated for all twenty-nine goals and resulted in two or more sub issues for every one of the goals. Looking once more at the example attribute *completeness*, the

questions formulated on the second GQM level have helped to identify the sub issues mentioned earlier such as maximum process completeness, responsibilities and monitoring procedures.

Addressing the final level of the GQM approach, the quantitative level (Metrics) was less straightforward than the two top levels. The difficulties that arose in addressing this level of the approach are addressed in the next section.

### 7.3 Items

Two or more sub issues have been identified for every quality attribute from the sets of concerns associated with the different viewpoints. The next step is to quantify these sub issues. Basili et al. [3] identify two types of metrics on the third (Metric) level of their GQM approach, introduced in the previous section. The first type is the objective metric. This metric relies purely on the object under measurement and not the perception through one or more viewpoints. An example of an objective metric would be the number of architects involved in the development of one architecture products or the amount of man-hours spent during this process. The second type of metric is the subjective metric. This metric takes the perspective of relevant viewpoints into account [3]. Subjective metrics ask opinions of respondents about the object under investigation. An example would be to answer the questions: 'Does the development process always realize timely delivery of its resulting products?' and 'Is there sufficient information available to assess the performance of the process with respect to timely delivery?'

In realizing an objective quality instrument objective metrics would be preferable. The problem with this of type metric in the context of the enterprise architecture development process is the lack of benchmark data. In order for an objective metric to be of any value there needs to be a scale available that includes a lower and an upper bound. Thus creating a common idea of what objective and quantifiable process properties make up a good process. However, in the field of enterprise architecture this data is not available and may not become available for some time. And gathering this information would require extensive benchmark studies and probably a more mature field of enterprise architecture. For that reason the choice was made to use perceptual and thus subjective measurements.

Based on the sub issues that were identified on the second level of the GQM approach, items were formulated. An item is a statement that represents the 'optimal' state of being for that item in relation to its parent attribute. A qualitatively sound development process would have to fulfill this statement. An iterative process of formulating and reformulating these items resulted in a total of sixty-nine items addressing the sub issues of all fifteen quality attributes. To illustrate this with an example, take a look at the quality attribute *completeness* once more. Table 13 shows the items that were created based on the sub issues were identified for the quality attribute *completeness* using GQM. These items address the sub issues associated with the viewpoints that have *completeness* as part of their set of concerns. Statement 1 addresses the level of process completeness, statements 2 and 4 address accountability issues, statement 3 and 5 go to the requirements, statement 6 looks at the usage of existing architecture products stored in a repository and finally statements 7 and 8 address governance issues. The table also lists the viewpoints that are associated with the items. As can be seen from this example, an item might be associated with one or more viewpoints, but every item belongs to at least one viewpoint. Appendix V gives an overview of all attributes, their corresponding items and the viewpoints these items belong to. In formulating these items an attempt was made to ensure objectivity. This included multiple peer reviews in order to remove room for interpretation errors and defining the items to represent the optimal state of being for each item and thus the corresponding quality attribute. The level of objectivity of the quality instrument will be further discussed in a later section.

Statement	Viewpoint(s)
Completeness	
1. The development process strives demonstrable to address all requirements in the architecture descriptions.	Customer, Executor, Governor, Owner
2. The responsibilities for realizing completeness are recorded before the development process is initiated.	Customer, Executor, Governor, Owner
3. The executing architects have a complete set of requirements at their disposal when the development process is initiated.	Executor
4. When the development process has resulted in incomplete architecture descriptions the architects responsible are held accountable.	Customer, Executor, Governor, Owner
5. Before the development process is initiated, the executing architects are supplied with a set of requirements that guarantee consistency with the existing collection of architecture descriptions.	Maintainer
6. During the development process architects also make use of existing architecture descriptions to address the requirements.	Maintainer
7. The development process includes procedures to monitor and report on its completeness.	Governor, Owner
8. There is sufficient information available to asses the completeness of the development process.	Governor, Owner

Table 13: The attribute completeness and its items

## 7.4 Scales

With the items of the quality instrument in place, the last step in creating the quality instrument was to define the scales on which to measure these items. Gitem et al. [18] state that: '*to quantify constructs that are not directly measurable multiple-item scales are often used*'. They advocate the use of multi-item scales over single item measures because multi-item scales have a higher reliability than single-item measurement. The Likert scale is one of the most commonly used multi-item scales, that provides respondents with the possibility to answer questions on a five-point scales from *Strongly Disagree (1)* to *Strongly Agree (5)*.

The choice was made to use a customized five-point Likert scale as a quantification of the items of this quality instrument. This customized five-point Likert scale varies from *Strongly Disagree (1)* to *Strongly Agree (4)* and omits the *Neutral* answer. The fifth point on the scale is made up by the *Not Applicable (0)* answer. This point was included to offer the architects the option to discard an item if this item is not applicable for their respective development process. The choice for not adding the *Neutral* answer was made deliberately to force respondents to make a choice, instead of taking the easy way out by providing the *Neutral* answer.

## 8 VALIDATION

---

### 8.1 Introduction

The previous chapter has described the creation of the quality instrument. And although the construction of this instrument has been realized in a structured manner based on existing literature, it still needed to be statistically validated to prove its reliability and validity. This chapter describes this validation process and the changes that resulted from it. The items and their corresponding scales were brought together in a questionnaire that was targeted at experienced architect in the Dutch business world. The population of respondents is described in section 8.2. In the questionnaire the concepts used were explained, but there was no description of the underlying framework provided. It consisted of all sixty-nine items, grouped per quality attribute, and their corresponding scales, as described in section 7.4. The questionnaire was pre-tested by transmitting it to three respondents that were not included in the final population of respondents. The primary focus of these pre-tests was to test the clarity of the items, and locate problems with multi-interpretability or the structure of the questionnaire. Problems that were identified during this pre-test phase were addressed before sending out the questionnaire. The responses to the questionnaire formed the input for the cross-sectional validation of the quality instrument. This validation process is described in section 8.3. Section 8.4 summarizes the results of the validation of the instrument and discusses some additional insights that were gathered during the process.

### 8.2 Population

The questionnaire was transmitted to a total of eighty possible respondents. All of these respondents are (or were) active in the field of (enterprise) architecture and are working at different companies throughout the Netherlands. In an attempt to consolidate their cooperation total anonymity of their responses was guaranteed. As a quid pro quo for their efforts a benchmark report was promised to them relating their personal results to the results of the total population. After an initial period of waiting, respondents were send reminders via email or were contacted in person and by telephone. These efforts resulted in a total of twenty-six responses. Table 4 describes the most important features of the participating members and their corresponding organizations. The responses were carefully examined to verify that all items were answered correctly. This led to the deletion of one of the responses, resulting in a sample size of twenty-five valid respondents divided over twenty-three different companies ranging in size from six hundred to ninety-seven thousand employees worldwide.

<b>Field</b>	<b># respondents</b>	<b># employees</b>
Finance	6	1100 – 97000 (worldwide)
Telecom	3	600 – 31116
IT service providers	3	3612 – 31116 (worldwide)
Other	7	1000 – 90000 (worldwide)
Medicine	2	6500
Insurance	2	2000 – 7000
Government	3	30000 – 37000

*Table 14: Population details (2004)*

### 8.3 Statistical analysis

In the statistical analysis of the quality instrument, the example of Flynn et al. [16], Straub [35], Nagpaul [27] and Adcock et al. [1] was followed. These authors distinguish two distinct properties to be addressed in validating a quality instrument. The first property is reliability of the instrument. The reliability will be discussed in section 8.3.1. The second property is validity that is discussed in section 8.3.2.

#### 8.3.1 Reliability

The data set resulting from the 25 valid responses provided the input for validating the quality instrument. The first step in this validation process was determining the reliability of the instrument, its different quality attributes and their related items. Peterson [30] and Flynn [16] describe reliability of an instrument as its ability to yield consistent measurements. In other words: '*a reliable instrument enables respondents to answer (approximately) the same questions the same way every time*' [32]. Peterson identifies two types of reliability coefficients, those based on longitudinal data (for example: test-retest reliability coefficient) and those based on cross-sectional data (for example: internal consistency reliability coefficients and equivalence reliability coefficients). Due to time constraints of both the respondents and this thesis project, the longitudinal approach using test-retest coefficients was excluded as a realistic opinion. The cross-sectional validation approach provided a more feasible opinion. And the data resulting from the questionnaire would be highly suitable for this type of reliability analysis.

The most commonly used reliability coefficient for cross-sectional data analysis is Cronbach's alpha. Cronbach's alpha is: '*a generalized measure of the internal consistency of a multi-item scale*' [30]. It determines whether a set of items measures the same construct by calculating the inter-item correlation of these items. The quality instrument consists of fifteen scales (the quality attributes), each of the scales represented by two or more items. Thus making it into a multi-scale instrument and suitable for adopting Cronbach's alpha in determining its reliability. Cronbach's alpha provides a degree of reliability on a scale from zero to one, zero representing totally unreliable and one totally reliable. Existing literature does not provide an unambiguous answer on what constitutes an acceptable lower bound for the reliability of a quality instrument. Peterson [30] also acknowledged this lack of guidance and performed empirical research on the lower bounds for reliability alpha that can be found in scientific literature. His research shows that an alpha of 0.7 as a lower bound for reliability was observed in seventy-five percent of the cases examined. However he stipulates that the reliability of an instrument should be viewed as a function of its research purpose. Exploratory research with newly formed scales should adopt a lower minimum alpha than for example applied research [30]. Flynn et al. [16] adopt a similar vision in determining the reliability of their newly developed quality management instrument for manufacturing companies. In validating their instrument they adopted a lower bound for reliability of  $\alpha=0.6$ . Their choice for this lower bound was based on the fact that their instrument was composed entirely of new scales and contained scales with a small number of items. The same reasoning can be adopted in relation to the quality instrument that has been developed in this thesis for the enterprise architecture development process. This instrument is the first step in quality management for the enterprise architecture process. The scales that have been defined are completely new and some of them consist of a small number of items. For that reason a lower bound of  $\alpha=0.6$  was adopted as acceptable degree of reliability for this quality instrument. Table 15 describes the actions to be undertaken based on the Cronbach's alpha calculated for the different scales of the instrument. Alpha's  $> 0.7$  will be accepted without further actions.

And although the choice was made to accept scales with  $\alpha > 0.6$ , the scales scoring between 0.6 and 0.7 would be analyzed further to determine whether they could be improved by deleting one or more of the items in a scale. This choice shows the ambition to create an instrument that meets the more established lower bound of 0.7. This decision should be made with the impact on content of that scale. If deletion of an item results in an incomplete scale, the respective item would not be removed from the scale. Scales with an  $\alpha < 0.6$  will need to be analyzed more extensively. If deletion of one of its items does not result in an  $\alpha > 0.6$ , the entire scale will be deleted from the instrument.

Cronbach's alpha:	Action:
$\alpha < 0.6$	Further analysis to determine whether deletion of items results in a sufficiently high Alpha.
$0.6 < \alpha < 0.7$	Further analysis to determine whether deletion of items results in a higher Alpha.
$\alpha > 0.7$	Accept without further action

Table 15: Cronbach's alpha acceptance levels

### 8.3.1.1 Results

For calculating the reliability coefficients, the software package SPSS was used. The data set resulting from the questionnaire was checked for errors and all *Not Applicable* answers were deleted from the data set. Although these answers did provide insight into the development process of the respective respondent, they did not go to the content of the items. Certain items or even an entire scale might not be applicable for a certain architecture development process. An example can be found in the approach DYA<sup>®</sup>. This approach to enterprise architecture advocates not maintaining a repository of architecture products. This would result in all items related to the *maintainer* viewpoint being answered *Not Applicable* and thus scoring zero. To prevent these answers from influencing quality score in a negative fashion, these items were deleted from the data set. However, these answers were not disregarded completely and will be discussed further in section 8.4.

The calculation of the Cronbach's alpha resulted in twelve scales with an alpha between 0.717 and 0.917. These scales were accepted without changes. The results of these calculations can be viewed in table 16. The remaining three scales needed some further attention. The first scale that resulted in an  $\alpha < 0.7$  was the quality attribute *analyzability*. The initial alpha of *analyzability* was 0.471. Further analyzes of the scale pointed at the third item in the scale as a candidate for removal from the scale. Its removal resulted in an alpha of 0.718, making *analyzability* into a reliable scale. Originally the scale *analyzability* consisted of three items. Deletion of this item reduced the total of items to two. Flynn et al. [16] advocate maintaining at least three items in each scale to ensure good representation. But since there were two scale that started out with but two items, the scales *understandability* and *learnability*, this restriction will not be imposed in this thesis project. Scales with but one item remaining are not acceptable, these will be deleted from the instrument entirely. The second scale with an  $\alpha < 0.7$  was the two-item scale *understandability*. Because this scale consisted of two items from the beginning, the removal of one of the item was not an option. For that reason the scale was removed completely from the instrument. The third scale with an  $\alpha < 0.7$  was *learnability*. With an alpha of 0.578, the lower bound of 0.6 was not met either. And since this too was a two-item scale, it was also removed from the quality instrument.

Table 6 shows the results of the calculations and subsequent removals for every scale, it shows the number of the respondents incorporated in the calculations, the resulting Cronbach's alpha, the number of items remaining in the scale after removal and the number of items that were removed based on the calculations. Following table 6 the scales for *understandability* and *learnability* should be removed from the scale. Table 6 reflects these decisions.

	<b>Number of respondents</b>	<b>Cronbach's alpha</b>	<b>Number of items in scale</b>	<b>Number of items deleted</b>
<b>Dimension I: Functionality</b>				
Timeliness	25	0.824	6	0
Responsiveness	23	0.717	3	0
Completeness	22	0.773	9	0
Suitability	25	0.835	6	0
Consistency	21	0.913	8	0
Inter-operability	24	0.821	3	0
Repeatability	23	0.901	5	0
<b>Dimension II: Usability</b>				
Understandability	25	0.462	0	2
Learnability	24	0.578	0	2
Operability	22	0.876	3	0
<b>Dimension III: Maintainability</b>				
Analyzability	22	0.718	2	1
Modifiability	24	0.878	3	0
<b>Dimension IV: Reliability</b>				
Failure frequency	24	0.917	5	0
<b>Dimension V: Efficiency</b>				
Time behavior	24	0.806	3	0
Resource behavior	24	0.869	3	0

Table 16: Results of the reliability study

But with the scales *understandability* and *learnability* removed from the quality instrument, the dimension *usability* would be brought back from three to just one scale. A closer look at the items of both scales showed that both are two-item scales. The items of the two scales address similar sub issues, namely the availability of study materials and the effort required from architects. The final decision with regards to these items will be discussed further in sections 8.4.

### 8.3.2 Validity

Flynn et al. [16], Straub [35], Nagpaul [27] and Adcock et al. [1] identify instrument validity as the second important property for a quality instrument. Together they identify four different types of validity. Two of these have been examined for the quality instrument for the development process. The following sections address these two types of validity, content and construct validity.

#### **Content validity:**

Content validity has been defined as: '*the adequacy with which a specific domain has been sampled*' [28]. Straub [35] addresses this type of validity as being '*difficult to achieve and even more difficult to verify*'. An important reason for this is the fact that its verification is judgmental and subjective.

Literature reviews uncovered five key issues in verifying an instrument's content validity. The first issue addresses the degree to which the collection of items is representative for the domain [16]. The second issue addresses the sensibility of the methods that were used in constructing the instrument [16, 28]. The third issue relevant to the content validity of a quality instrument takes into account whether there were key items omitted from the instrument scales [1]. This author also provided the fourth issue, namely whether inappropriate items were included in the scales. The final issue in verifying content validity is closely related to the first and second issue, but worth mentioning explicitly, namely the quality of the literature study undertaken in constructing the instrument. As was mentioned earlier the verification of content validity of an instrument is judgmental and subjective. However, if the five issues mentioned above have been taken into account in the construction of the quality instrument a case can be made for the content validity of the quality instrument.

The construction of the quality instrument has been described extensively in the preceding chapters. The structured approach that started by defining the terminology, followed by the creation of a theoretical framework that formed the foundation for the quality instrument, goes to the content validity of the instrument. Extensive literature reviews formed the theoretical backbone for all steps undertaken in the construction of the framework and answers questions about the quality of the literature study and whether instrument items are representative for the domain. As a result of the repeated literature reviews, theory from more established fields such as software engineering and business process management was employed in the development of the quality instrument. A structured approach in the form of GQM was employed in identifying the sub issues for the different quality attributes. This approach addresses the issues of redundancy and omission of relevant items. In conclusion it can be stated that the structured approach described in the preceding chapter implies an acceptable level of content validity for the quality instrument for the enterprise architecture development process.

**Construct validity:**

The second form of instrument validity is construct validity. Straub defined construct validity as: '*whether the measures chosen are true constructs describing the event*' [35]. In other words, construct validity is: '*the extent to which the items in a scale all measure the same construct*' [16]. Again following the example of Flynn et al. [16] the construct validity was calculated using factor analysis, in the form of principal component analysis. The items in a scale were formulated to represent the sub issues of the underlying quality attribute. Factor analysis provides a way to examine to what degree the items in a scale capture its underlying concept. In factor analysis this concept is called a factor and the items belonging to one factor are correlated higher to each other than to item of another factor. In the validation of the quality instrument for the architecture development process, the scales represent the factors. To calculate the construct validity of this instrument, the factor analysis variant principal component analysis was used to determine whether the items of a scale all load one component (factor), the scale in question.

### 8.3.3 Results

Each of the scales was analyzed separately. Initially the principal component analysis identified four scales that loaded on more than one component. These four scales, *timeliness*, *completeness*, *suitability* and *consistency*, were analyzed further on possible improvements. The scale *timeliness* originally consisted of six items. The factor analysis pointed out that one item, that addressed the sub issue of accountability, did not measure on the same construct as the other five items. After deletion of this item from its scale, the remaining five items loaded on the same component. *Consistency* also contained one item related to the sub issue of accountability that did not load on the same component as the other items in the scale did. For that reason one item was deleted. The same situation occurred for the scale *suitability*. This resulted in the deletion of one item from this scale. And again this item handled about the sub issue accountability. The scale *completeness* needed more attention. Initially this scale loaded on three different components. After the deletion of two items, the remaining items in the scale all loaded on the same component and thus all measured the same construct. As a result two items relating to the sub issues of maximum completeness and accountability were removed from the instrument. There is a remarkable similarity noticeable between the items that were deleted due to the principal components analysis. All but one of the removed items addressed the sub issue of accountability for the respective scales. More precisely, the recording of accountability before the development process was initiated. This might indicate that there is a structural problem with either the formulation of these items or the recording of accountability at the organizations of the respondents. However, since the results of the principal component analysis dictate the removal of these items, they were removed from the quality instrument. These decisions are discussed further in section 8.4.

As a result of the actions described above, all items in the different scales loaded on one component, measuring the same construct. The Eigenvalues of all these scale remained well above the minimum criterion value 1. For the scales *timeliness*, *consistency*, *completeness* and *suitability* the Cronbach's alpha's were recalculated and still found to be above the lower bound of  $\alpha=0.7$ . Table 17 shows the results of the factor analysis. The first column shows the Eigenvalues of the different scales, the second column ('A') shows the new Cronbach's alpha after the deletion of the items and the last set of columns show the loadings of the remaining items on the remaining scale.

Scale	Eigen-value	A	Factor loadings						
			Item #1	Item #2	Item #3	Item #4	Item #5	Item #6	Item #7
<b>Functionality:</b>									
Timeliness	3.107	0.839	0.664	0.829	0.864	0.756	0.812		
Responsiveness	2.009	0.717	0.804	0.747	0.897				
Completeness	3.122	0.798	0.653	0.698	0.591	0.606	0.873	0.854	
Suitability	3.015	0.834	0.639	0.716	0.725	0.892	0.878		
Consistency	4.793	0.918	0.787	0.840	0.861	0.764	0.660	0.898	0.949
Inter-operability	2.271	0.821	0.893	0.905	0.810				
Repeatability	3.598	0.901	0.917	0.834	0.889	0.802	0.792		
<b>Usability:</b>									
Understandability	1.303	0.465	0.807	0.807					
Learnability	1.410	0.578	0.840	0.840					
Operability	2.410	0.876	0.938	0.904	0.844				
<b>Maintainability:</b>									
Analyzability	1.571	0.718	0.886	0.886					
Modifiability	2.423	0.878	0.959	0.783	0.945				
<b>Reliability:</b>									
Failure frequency	3.805	0.917	0.872	0.863	0.782	0.908	0.929		
<b>Efficiency:</b>									
Time behavior	2.161	0.806	0.769	0.925	0.845				
Resource behavior	2.429	0.869	0.933	0.911	0.853				

Table 17: Results of the factor analysis

**Other forms of validity:**

The authors mentioned before [13, 32] acknowledge two other types of instrument validity. Flynn et al. [16] addresses criterion-related validity, a type of validity that determines the correlation between a set of objective quantitative measurements and the results of the questionnaire. However, since there are no such objective quantitative measurements available for the enterprise architecture development process, this type of validity can not be determined for this quality instrument. Straub [35] describes statistical conclusion validity in which mathematical relationships between variables are examined. This type of validity is strongly influenced by the size of the population and only provides valuable results if a wide range of other environment variables with respect to reliability and validity have been met. The relatively small population and the fact that this thesis is an exploratory research into the quality of the enterprise architecture process were reason for not examining the instrument's statistical conclusion validity.

## 8.4 Summary

### 8.4.1 The instrument

The validation of the quality instrument has resulted in a quality instrument for the pre-defined development process that is both reliable and valid. The reliability of the instrument has been analyzed using Cronbach's alpha and has resulted in the deletion of one item for the scale *analyzability*. The scales *understandability* and *learnability* should be removed entirely based on their respective Cronbach's alphas. However, as was discussed earlier there are remarkable resemblances noticeable between the items of both removed scales. Both scales consist of but two items and the items of both scales address the same sub issues. These similarities indicate a possible structural problem with two-item scales and/or the formulation of their respective items. This thesis project has not provided a satisfactory answer to this problem. Extended research might identify the actual reason behind these anomalies, enabling the scales to be added in the quality instrument once more. However, based on their Cronbach's alphas these scales were removed from the quality instrument. The remaining scales all exceed the lower bound of  $\alpha=0.7$  for instrument reliability, thus proving the instrument reliable. The validity of the instrument has been verified on two different levels, content and construct validity. The key issues for content validity were listed and addressed to substantiate an acceptable level of content validity for the quality instrument. The second level of validity, construct validity, was analyzed using principal component analysis. This analysis resulted in the removal of another five items, one for the scale *timeliness*, one for *consistency*, one for *suitability* and two for the scale *completeness*. Again a remarkable resemblance was found between the removed items. Four out of five items were related to the sub issue of accountability. And although this research did not provide satisfactory answer to explain these anomalies, future research might uncover those answers. As a result the instrument now consists of fifty-five items and is proven to be both reliable and valid. The resulting scales and their corresponding items can be viewed in Appendix V.

### 8.4.2 Gathered insights

#### *Completeness*

As was discussed in section 3.5 the quality instrument is not complete in the sense that it addresses all quality attributes that have some relevance to the quality of the enterprise architecture development process. However, the attributes that were included in the instrument are well represented by a scale of two or more items. These items address the sub issues that were identified for the different quality attributes. Practical application of the instrument and additional research could provide extension of the instrument by adding more quality attributes to the instrument and improving the existing scales.

#### *Not Applicable answers*

As was discussed in section 8.3.1 the *Not Applicable* answers were deleted from the data set before the statistical analysis. This was done because these answers did not go to the contents of the respective item. The *Not Applicable* answers, represented by the value zero, would have a negative influence on the scores of the respondents, thus provide a distorted view of their process quality and result in incorrect reliability and validity calculations. But removing the *Not Applicable* answers from the data set does not imply that these answers should be ignored completely. Chapter 9 addresses this issue more elaborately.

### *Qualitative results*

Next to the quantitative results, the questionnaire also provided the answer to a number of qualitative questions about the scales and their respective items. These qualitative results pointed to the quality attributes *complexity*, *generality* and *progress monitoring* as being the first attributes eligible for future inclusion in the instrument. The qualitative results also answered a number of questions about the usability of the instrument itself. On average it took respondents between half an hour and one hour to address all items. And although the quality instrument will most likely not be applied in questionnaire form in the future, this does provide a promising indication of the usability of the instrument. The application of the instrument will be addressed more elaborately later in chapter 9. Last, the respondents generally agreed that the items were readable and understandable with few problems due to different interpretations.

### *Maturity*

A final point that became clear during the validation process described in the previous sections was that the developed instrument does presume some level of architecture maturity within the organization. Some form of an architecture process needs to be in place in order for this instrument to be applicable. This process needs to be executed at least once for architects to be able to perform the quality analysis by means of this quality instrument.



## 9 APPLICATION

---

### 9.1 Introduction

The main goal of this thesis project was to develop a scientifically validated quality instrument for the enterprise architecture development process. The previous chapters have described the approach that was followed in achieving this goal. With a scientifically validated instrument in place, there remains one last issue to be addressed and that is the application of the instrument. Although the practical application of the instrument was not the subject of extensive research, this thesis project has resulted in some important insights that will be discussed in the following sections.

### 9.2 Type of instrument

The quality instrument can be applied as an instrument for scientific quantitative research in the form of benchmark studies into the quality of enterprise architecture development processes. This type of research could provide a valuable contribution to the maturing field of enterprise architecture by examining relationships between for example quality attributes, process quality and maturity, and the process quality of organizations in similar fields of expertise or similar size. These are just a few examples of possible research subjects for benchmark studies. The second possible application of the quality instrument is a more practical one. Organizations could employ the instrument to measure the quality of their current architecture development process. Strengths and weaknesses of these processes can be identified using this instrument, providing tangible handles for process improvement. A standard and generally applicable instrument, such as the one developed during this thesis project, also makes it possible for companies to compare practices.

The qualitative results of the validation process described in chapter 8 showed that a questionnaire is not the most suitable format for deploying the quality instrument. This type of deployment does not provide respondents with the opportunity to ask for explanations of the concepts used, the theoretical framework or the instrument items. Thus, creating possible interpretation errors and as a result inaccurate quality measurements. These issues can for the most part be solved by deploying the instrument as an auditor tool. The auditor would be available to provide explanations where needed. A second advantage would be that the auditor can make sure that the right stakeholders are approached to execute the instrument. By having the different stakeholders answer the items for the viewpoint they are associated with, the instrument is deployed most optimally and will result in a representative measure for process quality. This type of application will be discussed further in the next section. A third important role of the auditor would be in interpreting the resulting measurements. This will be discussed further in section 9.4.

### 9.3 Application

The validation of the quality instrument has resulted in a reliable and valid instrument consisting of fifty-five items. These items represent the perception of quality of the development process for its five viewpoints. The application of the instrument should take these different viewpoints into account. This can be realized by providing the stakeholders of a certain viewpoint only with the selection of items belonging to that viewpoint. To illustrate this, look again the items that make up the scale for the quality attribute *completeness*, table 13. Stakeholders represented by the *customer* viewpoint would be asked to answer items 1, 2 and 4. Likewise the stakeholders represented by the other viewpoints would be asked their associated items. By applying the instrument in this fashion, the perceptions of quality from the different viewpoints is included in the

measurement. This way a maximal representative measurement of the development process quality is realized. To realize this instrument application the fifty-five items of the quality instrument would be split up in five subsets of items, each subset containing the items for a distinct viewpoint.

## 9.4 Results

Figure 11 shows the average results of all respondents that participated in the second questionnaire [41]. The first bar represents the average of fifty-two percent over all thirteen quality attributes. A similar representation would result from looking at one single enterprise architecture development process. As described in the previous sections, the quality instrument is best applied as an audit tool. The auditor would play an important part in calculating and interpreting the results as they are shown in figure 11. An key part of the auditor's work consists of handling the *Not Applicable* answers. These answers are not included in calculating the scores on the different quality attributes, but they were included in the instrument for a reason. So what could be the reason behind a *Not Applicable* answer? Does the sub issue associated with this item form a possibility for improvement or is this sub issue not applicable to the process due to reasons of maturity, complexity or process structuring? The auditor should be able to provide the answer to this question. For example, DYA<sup>®</sup> introduces the principle of providing just enough architecture at just the right time and does not advocate maintaining a repository of architecture products. As a result a process setup according to the DYA<sup>®</sup> principles would score *Not Applicable* on the items relating to the *maintenance* viewpoint. On the other hand, *Not Applicable* answers might also indicate shortcomings to the development process and indicate possibilities for immediate improvements. It would be the responsibility of the auditor to identify this as the reason behind these answers.

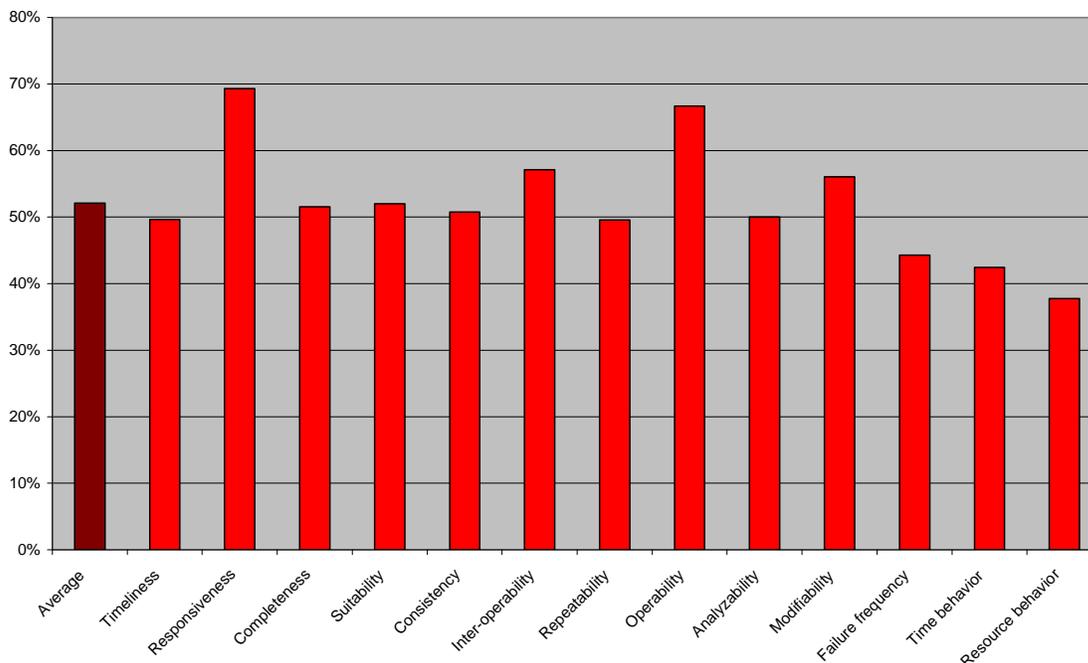


Figure 11: Benchmark results of all respondents over all attributes

The auditor is also responsible for interpreting the results as they are shown in figure 11. Looking at this figure the attributes *responsiveness* and *operability* stand out because of significant above average scores. *Failure frequency*, *Time* and *Resource behavior* on the other hand score significantly below average. An in-depth analysis of the items corresponding to the respective attributes can provide the auditor with tangible handles for process improvement. Figure 12 shows an in-depth analysis of the quality attributes *completeness* and *consistency*. It depicts for both attributes the average scores of the items relating to the different viewpoints associated to these viewpoints. Every bar in figure 11 represents the score belonging to one of the viewpoints. The bar most right shows the average score over all five viewpoints. Taking a closer look at *completeness*, it can be seen in figure 12 that the items relating to the *customer* viewpoint score significantly lower than those relating to the other viewpoints. With an average of thirty-nine percent it also scores below the average of fifty-two percent. If this had been the score of an individual architecture development process, the item(s) relating to this viewpoint would have to be evaluated to identify possibilities for process improvement. From the *executor* perspective the attribute *completeness* scores much better. Again in an individual case, a more in depth examination of the corresponding instrument items might uncover reasons for this above average score. The second quality attribute depicted in figure 12 shows a more even distribution over all five process viewpoints. Only the *customer* viewpoint shows a score significantly above average. It would be interesting to investigate the corresponding instrument items to discover why the process quality issues related to *consistency* score higher for the *customer* viewpoint.

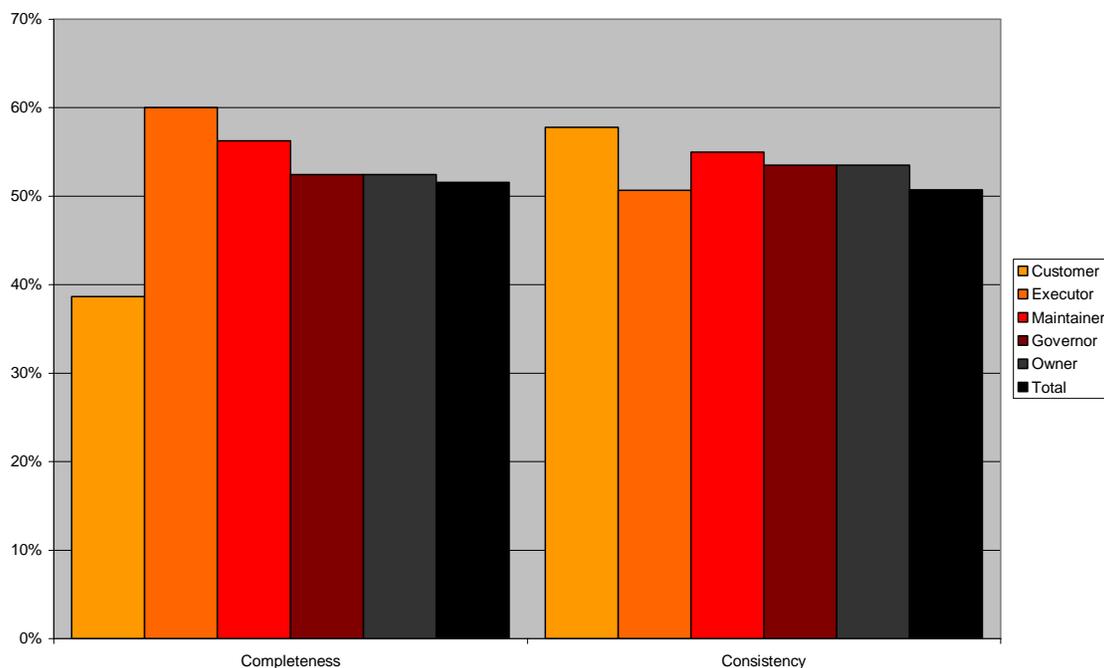


Figure 12: Example of the attributes completeness and consistency

Graphical representation as it is shown in figures 11 and 12 is very useful in providing a clear understanding of the quality of the enterprise architecture development process. Its strengths and weaknesses can be identified immediately. However, identifying the causes behind these scores takes some more effort. The items, corresponding to the different viewpoints and their respective quality attributes need to be analyzed to identify possibilities for improvement or locate best-practice examples. It is the responsibility of the auditor to identify these items and the sub issue(s) associated with them and formulate recommendations for improving architecture process quality based on these sub issues.



---

## Part III: Conclusions and Discussion

The last part of this thesis presents the conclusion of this research in chapter 10. Chapter 11 discusses these conclusions and identifies promising subjects for future research.

---



## 10 CONCLUSIONS

---

This thesis project has provided the field of enterprise architecture with a valuable scientific contribution into the quality of enterprise architecture. The research question that guided this research was defined as:

“How can a scientifically validated method for measuring quality of the process of enterprise architecture be created?”

The answer to (part of) this question has been described in this thesis document. It describes the subsequent steps undertaken in the creation of a quality instrument for the enterprise architecture development process. The main contribution of this thesis project is the resulting quality instrument for the enterprise architecture development process. The reliability and validity were shown by means of statistical analysis. As such it provides organizations with a reliable and objective instrument for measuring the quality of their architecture development processes. This instrument is also a valuable addition to science in that it offers a tool for benchmark studies into the subject of process quality. However, the answer to the main research question has been given in the form of the structured approach that was adopted in developing the resulting quality instrument for the architecture development process and is described in this thesis document. This approach can be adopted by other researchers in the development of new instruments or for architecture practitioners in the expansion of the existing instrument.

The following sections describe the instrument development approach followed in this thesis project in more detail. This approach can be divided into three parts: the terminology, the development approach and the resulting instrument.

### **Terminology**

Enterprise architecture, architecture processes and products, and architecture process viewpoints are all concepts that are subject to continuous discussion. For that reason the first step in this thesis project consisted of defining these concepts. By establishing working definitions for the concepts used in this thesis, further discussion was avoided and the instrument development could commence. The key concepts that were defined included the enterprise architecture process and its sub processes and the viewpoint definition with its corresponding representation. The process definition was important in defining the scope of the project and the viewpoint representations would form the theoretical framework.

### **Instrument development**

With the central concepts defined, the first step in creating a theoretical foundation for the quality instrument was creating a quality model for the enterprise architecture development process. Adjoining disciplines such as software engineering and business process management provide the input for creating this model. This resulted in thirty-three identified quality attributes subdivided in 7 quality dimension, each attribute addressing its own characteristic of the architecture development process. The process viewpoints and their representations formed the second part of the theoretical framework. Five viewpoints on the development process were identified together with their respective stakeholders and concerns with respect to the quality of this process. These concerns were represented by a selection of quality attributes from the quality model. To assure a workable solution only the five to seven most relevant to the different viewpoint attributes were included in the viewpoint representations. The viewpoints and their corresponding concerns (quality attributes), have been summarized in table 18.

Customer	Executor	Maintainer	Governor	Owner
Timeliness	Completeness	Completeness	Timeliness	Responsiveness
Responsiveness	Suitability	Consistency	Completeness	Completeness
Completeness	Consistency	Repeatability	Suitability	Suitability
Suitability	Operability	Analyzability	Consistency	Consistency
Consistency	Learnability	Modifiability	Repeatability	Failure frequency
Inter-operability	Understandability		Failure frequency	
			Resource behavior	
			Time behavior	

Table 18: Summary of the theoretical framework

Based on the viewpoint representations partly shown above, the quality instrument for the enterprise architecture process was developed. In subsequent steps, the sub issues relevant to every quality attribute were identified, items were formulated based on these sub issues and finally the instrument scales were defined. The resulting quality instrument consists of sixty-five items to be rated on a customized five-point Likert scale. This Likert scale ranges from *Strongly Disagree (1)* to *Strongly Agree (4)* that has the *Not Applicable (0)* answer as a fifth option.

With the quality instrument constructed the next step in this project consisted of its validation. Following the example of a number of different authors [1, 16, 35], the reliability and validity of the quality instrument were analyzed. The results from a questionnaire, transmitted to architects working at different companies throughout the Netherlands, were used in cross-sectional validation. This questionnaire was composed of the sixty-five instrument items. The respondents were asked to apply the instrument to their own architecture development process. A total of twenty-five valid responses was received. The reliability of the quality instrument was examined using the commonly used reliability coefficient, Cronbach's alpha. After the removal of two unreliable scales and one unreliable item, the remaining instrument items all surpassed the minimum criterion for reliability, proving the instrument to be reliable. The validity of the quality instrument was examined on two different levels. The first level, content validity, was difficult to realize and even more difficult to prove, but the structured approach of this thesis project in combination with extensive literature reviews and the usage of proven methods such as the GQM approach provide an acceptable level of confidence about the content validity of the quality instrument. The second level of instrument validity that was examined was construct validity. This type of validity was calculated using principal component factor analysis. Based on these calculations five more instrument items were removed to realize a valid instrument. Appendix III gives an overview of the resulting instrument after these validation steps.

The last chapter of part II addressed the recommended application of the quality instrument as an auditor tool. In this type of application an important part is played by the auditor. It is the responsibility of the auditor to avoid interpretation errors, to assure correct instrument usage, assist in correctly interpreting results and identify improvement opportunities based on these results.

### Quality instrument

The resulting quality instrument consists of a total of 55 items associated with 13 quality attributes belonging to five quality dimensions of the quality model for the enterprise architecture development process. This instrument is proven to be both reliable and valid.

## 11 DISCUSSION

---

The results of this thesis project provide a valuable addition to the field of enterprise architecture, but there are a number of issues that have to be addressed. This chapter will address these issues and provide recommendations for promising future research subjects.

The first issues go to the content of the quality instrument. First there is the issue of instrument completeness. It was explained in section 6.7 that the resulting instrument, due to the choice to include only the most relevant quality attributes in each set of viewpoint concerns, does not provide a complete measure for architecture development process quality. Future research could make an effort to extend the instrument with new scales. The qualitative results, described in section 8.4.2, show the quality attributes *complexity*, *generality* and *progress monitoring* as first candidates to be included as new instrument scales.

A more difficult issue arose during the statistical analysis of the instrument. In realizing a reliable instrument, two entire scales were deleted from the quality instrument. Both of these scales were part of the dimension *Usability*, reducing this dimension to one scale. A remarkable resemblance between these scales can be found in the fact that both started out as two-item scales. Taking a closer look at the items of these scales showed that the items of the respective scales address the same sub issues, namely available study materials and the effort required from architects. These similarities arouse suspicions about a structural problem with two-item scales and/or the formulation of the items. However, this research was unable to provide a suitable explanation for these issues. Future research and application of the instrument might provide more insight. Another opportunity for future research can be found in an alternative approach to analyzing the reliability of the quality instrument, a longitudinal reliability analysis. This type of test-retest approach would increase the confidence in the resulting instrument.

A similar problem was encountered in establishing the construct validity of the instrument. Factor analysis resulted in the removal of five items from four different scales. Four of these items addressed accountability sub issues and more precisely, the recording of responsibilities. This leads to one of two conclusions. Either accountability is a subject that left mostly unaddressed in the enterprise architecture development process or the formulation of the corresponding items was structurally wrong. Future research should provide the answer to which conclusion is the right one.

The last issue with regards to the instrument's contents goes to the objectivity of the instrument. The use of subjective, perceptual measurements does not provide a strictly objective quality instrument. However, during the formulation of the items and by recommending instrument application in the form of an audit instrument efforts were made to assure a certain level of instrument objectivity. Perhaps future research in the form of benchmark studies could help to provide objective quality measures to replace the subjective measures used in this instrument.

The instrument focuses on measuring the quality of enterprise architecture development as it was defined in chapter 3. The remaining part of the enterprise architecture process, the deployment process and its associated governance process, has been left outside the scope of this thesis project. Future research into the quality of this part of the architecture process could result in a quality instrument for the entire enterprise architecture process. In developing such an instrument a similar work plan could be adopted as was described in this thesis document.

Next to the research opportunities described above, there remain many more challenges in the field of enterprise architecture for both scientists and practitioners. Enterprise architecture is a relatively young field that does not identify any standard definitions for its main concepts or provide standard approaches towards working with architecture. Standardization of concepts and approaches are just two examples of subjects that help in maturing the field of enterprise architecture in which both science and practice could play important roles.

## 12 REFERENCES

---

- [1] Adcock, R., Collier, R. (2001). "Measurement Validity: A Shared Standard for Qualitative and Quantitative Research." *The American Political Science Review* 95(3): 529 - 546.
- [2] Anjard, R. P. (1996). "Process Mapping: One of Three, New, Special Quality Tools for Management, Quality and All Other Professionals." *Microelectronics Reliability* 26(2): 223 - 225(3).
- [3] Basili, V. R., Caldiera, G., Rombach, H.D. (1994). "The Goal Question Metric Approach." *Encyclopedia of Software Engineering* 1: 528 - 532.
- [4] Beamon, B.M., Ware, T.M. (1998) "A Process Quality Model for the Analysis Improvement, and Control of Supply Chain Systems." *Logistics Information Management* 11(2): 105-113
- [5] Berg, v. d., M.J.B.K., Steenbergen van, M.E. (2004). "DYA®: stap voor stap naar professionele enterprise-architectuur." TenHagenStam Uitgevers.
- [6] Bernus, P. (2003). "Enterprise models for enterprise architecture and ISO9000:2000." *Annual Reviews in Control* 27: 211-220.
- [7] Bevan, N. (1999) "Quality in Use: Meeting User Needs for Quality" *Journal of Systems and Software* 49(1): 89-96
- [8] Boeters, A., Noorman, B. (1997). "Kwaliteit op maat, IT-projecten met de RADAR-methode." Deventer, Kluwer Bedrijfswetenschappen.
- [9] Buchanan, R. (2002). "Evolving the Enterprise Information Architecture: Part 1 - Getting started." *Enterprise Planning & Architecture Strategies*, Meta Group. 6.
- [10] Cechich, A., Piattini, M., Vallecillo, A. (1998) "Component-Based Software Quality, Methods and Techniques." Springer-Verlag, Berlin.
- [11] Council, C. I. O. (2001). "A Practical Guide to CIO Council Federal Enterprise Architecture."
- [12] Clark, C. (1997). "Disease control programs should apply 20/80 rule." *Aids Weekly Plus*. 0(February 1997): 28
- [13] Emory, C.W. (1980). "Business Research Methods" Richard D. Irwin, Honeywood, IL, p.16
- [14] Finkelstein, A., Kramer, J., Nuseibeh, B., Finkelstein, L., Goedicke, M. (1992). "Viewpoints: A Framework for Integrating Multiple Perspectives in System Development" *International Journal of Software Engineering and Knowledge Engineering* 2(1): 31-58
- [15] Finkelstein, A., Sommerville, I. (1996) "The Viewpoints FAQ." *Software Engineering Journal* 11: 2-4
- [16] Flynn, B. B., Schroederb, R.G., Sakakibara, S (1994). "A framework for quality management research and an associated measurement instrument." *Journal of Operations Management* 11: 339-366.

- [17] Galin, D. (2004). *"Software Quality Assurance."* Harlow, Pearson Education Limited.
- [18] Gliem, J. A., Gliem, R.R. (2003). *"Calculating, Interpreting, and Reporting Cronbach's Alpha Reliability Coefficient for Likert-Type Scales."* 2003 Midwest Research to Practice Conference in Adult, Continuing and Community Education, Columbus, Ohio.
- [19] Harrington, H. J. (1991). *"Business Process Improvement: The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness."* New York, McGraw-Hill.
- [20] Hoogervorst, J. A. P. (2004). *"Enterprise Architecture: Enabling Integration, Agility and Change."* International Journal of Cooperative Information Systems 13(3): 213 - 233.
- [21] IEEE (2000). *"IEEE Recommended Practice for Architectural Description of Software-Intensive Systems."* New York, The Institute of Electrical and Electronics Engineers, Inc.
- [22] ISO 9000:2000 (2000) *"Quality management systems – Fundamentals and vocabulary."* ISO/TC176/SC2
- [23] Kimber, R.J., Grenier, R.W., Heldt, J.J. (1997) *"Quality Management Handbook"* Marcel Dekker Ltd.
- [24] Kraus, T.W., Büchler, M.W., Herfarth, C. (2005) *"Relationships between Volume, Efficiency, and Quality in Surgery – A Delicate Balance from Managerial Perspectives."* World Journal of Surgery 29(10): 1234-1240.
- [25] Mellis, W. (1998). *"Software quality management in turbulent times – are there alternatives to process oriented software quality management?"* Software Quality Journal 7(3/4): 277-295
- [26] Mullery, G. P. (1979). *"CORE - A Method For Controlled Requirement Specification."* The 4th international conference on Software engineering, IEEE Press.
- [27] Nagpaul, P. S. (1995). *"Quasi-Quantitative Measures of Research Performance: An Assessment of Construct Validity and Reliability."* Scientometrics 33(2): 169 - 185.
- [28] Nunnally, J. C. (1978). *"Psychometric Theory."* New York, McGraw-Hill.
- [29] Ortega, M., Pérez, M., Rojas, T. (2003). *"Construction of a Systematic Quality Model for Evaluating a Software Product."* Software Quality Control 11(3): 219 - 242.
- [30] Peterson, R. A. (1994). *"A Meta-analysis of Cronbach's Coefficient Alpha."* The Journal of Consumer Research 21(2): 381 - 391.
- [31] Rijsenbrij, D. (2005). *"Kanttekeningen bij 'Architectuur in de Digitale Wereld'."* version 0.6. Zeist.
- [32] Satpathy, M., Harrison, R., Snoolk, C., Butler, M. (2000). *"A Generic Model for Assessing Process Quality."* The 10th International Workshop on New Approaches in Software Measurement, Springer-Verlag.

- [33] Schekkerman, J. (2004). *"How to survive in the jungle of Enterprise Architecture Frameworks."* Victoria, Canada, Trafford.
- [34] Sommerville, I., Viller, S., Sawyer, P. (2001). *"Managing Process Inconsistencies Using Viewpoints."* IEEE Transactions on Software Engineering 25(6): 784 - 799.
- [35] Straub, D. W. (1989). *"Validating Instruments in MIS Research."* MIS Quarterly 13(2): 147 - 169.
- [36] The Open Group (2003). *"The Open Group Architecture Framework 'Enterprise Edition' version 8.1."*
- [37] Travis, P. (2005). *"Stand and Deliver."* Information Week (0)1027: 41
- [38] Ultsch, A. (2002). *"Proof of Pareto's 80/20 law and Precise Limits for ABC-Analysis."* Marburg/Lahn, University of Marburg.
- [39] Van den Bent, B. (2005). *"Workshop Quality and Architects."* Vianen
- [40] Van den Bent, B., Veltman - van Reekum, E. (2005). *"Personal interviews on the subject of EA quality."* Vianen
- [41] Van den Bent, B., Veltman - van Reekum, E. (2005). *"Questionnaire about the relevance of quality attributes in relation to process viewpoints."* Utrecht, Utrecht University.
- [42] Van den Bent, B., Veltman - van Reekum, E. (2005). *"Questionnaire to validate the quality instrument."* Utrecht, Utrecht University.
- [43] Van Zeist, B., Hendriks, P., Paulussen, R., Trienekens, J. (1996). *"Kwaliteit van softwareprodukten — Praktijkervaringen met een kwaliteitsmodel."* Deventer, Kluwer Bedrijfswetenschappen.
- [44] Veltman - van Reekum, E. (2005). *"Thesis on the subject of enterprise architecture quality."* Diemen, Utrecht, Utrecht University
- [45] Wagter, R., Van den Berg, M.J.B.K., Luijpers, J., Van Steenberghe, M.E. (2001). *"DYA® : snelheid en samenhang in business- en ICT-architectuur."* Den Bosch, Tutein Nolthenius.
- [46] Wijnstra, J. G. (2003). *"From Problem to Solution with Quality Attributes and Design Aspects."* The Journal of Systems and Software 66: 199 - 211.
- [47] Zachman, J. A. (1987). *"A framework for information systems architecture."* IBM Systems Journal 26(3): 454-470.

### **Appendix I: Definition of the Quality Attributes**

#### **Dimension Functionality:**

##### Compliance:

Indicators that bear on the degree to which the development process conforms to its own process model.

##### Completeness:

Indicators that bear on the degree to which the development process transforms all of the input requirements into the resulting enterprise architecture product(s).

##### Consistency:

Indicators that bear on the degree to which the development process does not introduce contradictions in the resulting enterprise architecture product(s).

##### Generality:

Indicators that bear on the ability of the development process to address conditions which are not covered by the input requirements but are relevant to the context in which the resulting enterprise architecture product(s) will be used.

##### Suitability:

Indicators that bear on the degree to which the development process matches the resulting enterprise architecture product(s) adequately to the input requirements.

##### Inter-operability:

Indicators that bear on the degree to which the development process contributes to the resulting enterprise architecture product's ability to function in its later environments.

##### Timeliness:

Indicators that bear on the degree to which the development process realizes the delivery of the resulting enterprise architecture product(s) within a specified range in time.

##### Responsiveness:

Indicators that bear on the degree to which the development process is designed to answer to service calls from its stakeholders.

##### Repeatability:

Indicators that bear on the ability of the development process (steps) to be repeated frequently and consistently.

#### **Dimension Scalability:**

##### Scalability:

Indicators that bear on the degree to which the development process maintains its efficiency in terms of time, cost and resource usage in handling a problem of larger dimension.

**Dimension Usability:**

## Understandability:

Indicators that bear on the effort required from an architect to understand the logical concepts of the development process.

## Learnability:

Indicators that bear on the effort required from an architect to learn to use the process.

## Operability:

Indicators that bear on the effort required from an architect to execute the development process with a level of confidence.

## Clarity:

Indicators that bear on the effort required to make an architect aware of the functions the development process can perform.

**Dimension Efficiency:**

## Time behavior:

Indicators that bear on the usage of time by the development process to realize the resulting enterprise architecture product(s).

## Resource behavior:

Indicators that bear on the usage of resources by the development process to realize the resulting enterprise architecture product(s).

## Complexity:

Indicators that bear on the ability of the development process to support various forms of complexity. (For example: complexity of input requirements and /or the resulting enterprise architecture product(s).)

## Process maturity:

Indicators that bear on the AMM (Architecture Maturity Model) level of the organization.

**Dimension Reliability:**

## Fault tolerance:

Indicators that bear on whether the development process can still continue in the presence of faults/failures in the process itself.

## Failure frequency:

Indicators that bear on the number of (and the interval) between failures encountered during the execution of the development process. The failures are due to defects in the design / implementation of the process itself.

## Recoverability:

Indicators that bear on whether the development process can attain its level of operation after a process failure has been addressed.

## Safety:

Indicators that bear on the degree to which the process addresses safety conditions which could handle the liveliness or the degraded performance and limit damage from failures.

### **Dimension Visibility & Control:**

Automatic checks & feedback:

Indicators that bear on the ability of the development process to provide feedback data and to support corrective actions if necessary.

Progress monitoring:

Indicators that bear on the ability of the development process to facilitate monitoring at any point of time during its execution to show that progress so far has been correct and effective.

Improvement measures:

Indicators that bear on the ability of the development process to support the analysis of the feedback data in combination with the data of previous runs and improve itself, or result in the improvement of a sibling process, continuously.

### **Dimension Maintainability:**

Analyzability:

Indicators that bear on the effort required from an architect to analyze the cause of a fault, process failure or unexpected feedback data.

Modifiability:

Indicators that bear on the effort required from an architect to address failures, detection of faults and unexpected data during the process execution; or faults discovered at a later time but linked to the process.

Stability:

Indicators that bear on the degree to which addressing a process fault adversely affect the development process itself. (Also: the frequency of changes done to the process.)

Testability:

Indicators that bear on the degree to which the development process could be validated itself.

Defect trend:

Indicators that bear on the trend of defects that are observed in the development process itself.

Formal verification:

Indicators that bear on the effort with which properties like consistency, correctness and invariant conditions of the enterprise architecture product under transformation could be formally verified during process execution.

Informal verification:

Indicators that bear on the effort with which properties like completeness, generality etc. of the enterprise architecture product under transformation could be informally verified during process execution.

Reusability:

Indicators that bear on the degree to which components of the development process can be reused in a different context.

Portability:

Indicators that bear on the degree to which the development process contributes to the portability of the resulting enterprise architecture product.

**Appendix II: Definitions of the Viewpoints on the Development Process**

<b>Name</b>	<b>Focus</b>	<b>Quality attributes</b>
Customer	The customer is the future user of the enterprise architecture product that results from the development process, in other words (one of) the executor(s) of the deployment process. The customer is also responsible for providing executing architects with part of the input requirements for the resulting product(s).	Timeliness, Responsiveness, Completeness, Suitability, Consistency, Inter-operability
Executor	The executor is the architect responsible for the creation of the enterprise architecture products. The executor transforms the requirements, constraints, etc into an enterprise architecture product taking into account the architectural principles.	Consistency, Understandability, Completeness, Suitability, Learnability, Operability,
Maintainer	The maintainer is responsible for maintaining an up-to-date and consistent repository of enterprise architecture products.	Consistency, Analyzability, Repeatability, Modifiability, Completeness
Governor	The governor is responsible for organizing and monitoring the enterprise architecture development process and all of its activities. The governor ensures that the process executed correctly and efficiently.	Completeness, Suitability, Timeliness, Failure frequency, Consistency, Repeatability, Time behavior, Resource behavior
Owner	The owner of the process is not directly involved in the enterprise architecture development process, but is responsible for the long term (tactical and strategic) impact of enterprise architecture and the business goals enterprise architecture needs to facilitate.	Responsiveness, Failure frequency, Consistency, Completeness, Suitability

### Appendix III: The Quality Instrument

Statement	Viewpoint(s)	Status <sup>a</sup>
Timeliness		
A fixed deadline for delivering the architecture descriptions is agreed upon before the development process is initiated.	Customer	M
The responsibilities for realizing timely delivery of the architecture descriptions are recorded before the development process is initiated.	Customer	D
The development process realizes delivery of the architecture descriptions within the fixed deadline.	Customer	M
When the deadline for delivering the architecture descriptions is not met, the architects responsible are held accountable.	Customer	M
The development process includes procedures to monitor and report on the timely delivery of the architecture descriptions.	Customer	M
There is sufficient information available to assess the performance of the development process on the about the timely delivery of architecture descriptions.	Customer	M

Statement	Viewpoint(s)	Status
Responsiveness		
Stakeholders have the possibility to influence the development process by means of change requests.	Customer, Owner	M
The development process is flexible enough to address change requests made by stakeholders in the architecture descriptions.	Customer, Owner	M
Architects always attempt to address change request in the architecture descriptions.	Customer, Owner	M

<sup>a</sup> This column shows a 'D' for items that were deleted from the quality instrument and 'M' for items that were maintained in the instrument after the statistical analysis.

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Completeness</b>		
The development process strives demonstrable to address all requirements in the architecture descriptions.	Customer, Executor, Governor, Owner	D
The responsibilities for realizing completeness are recorded before the development process is initiated.	Customer, Executor, Governor, Owner	D
The executing architects have a complete set of requirements at their disposal when the development process is initiated.	Executor	M
When the development process has resulted in incomplete architecture descriptions the architects responsible are held accountable.	Customer, Executor, Governor, Owner	M
Before the development process is initiated, the executing architects are supplied with a set of requirements that guarantee consistency with the existing collection of architecture descriptions.	Maintainer	M
During the development process architects also make use of existing architecture descriptions to address the requirements.	Maintainer	M
The development process includes procedures to monitor and report on its completeness.	Governor, Owner	M
There is sufficient information available to asses the completeness of the development process.	Governor, Owner	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Suitability</b>		
The development process leads to an adequate solution for all addressed requirements.	Customer, Executor, Governor, Owner	M
The responsibilities for realizing suitability are recorded before the development process is initiated.	Customer, Executor, Governor, Owner	D

The executing architects have a complete set of requirements at their disposal when the development process is initiated.	Executor	M
When the development process has resulted in unsuitable architecture descriptions the architects responsible are held accountable.	Customer, Executor, Governor, Owner	M
The development process includes procedures to monitor and report on its suitability.	Governor, Owner	M
There is sufficient information available to assess the suitability of the development process.	Governor, Owner	M

Statement	Viewpoint(s)	Status
<b>Consistency</b>		
The development process strives demonstrably not to introduce contradictions in the architecture descriptions between architecture descriptions.	Customer, Maintainer, Governor, Owner	M
The responsibilities for realizing consistency are recorded before the development process is initiated.	Customer, Executor, Maintainer, Governor, Owner	M
The operation procedure of the architects during the development process excludes the occurrence of contradictions in the architecture descriptions.	Executor	M
Contradictions in architecture descriptions that are discovered during or after the execution of the development process are addressed immediately.	Executor	D
All requirements that guarantee consistency with the collection of existing architecture descriptions are addressed in the architecture descriptions.	Maintainer	M
When the development process results in inconsistent architecture descriptions, the architects responsible are held accountable.	Customer, Executor, Maintainer, Governor, Owner	M
The development process includes procedures to monitor and report its consistency.	Governor, Owner	M
There is sufficient information available to assess the consistency of the development process.	Governor, Owner	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Inter-operability</b>		
The development process realizes architecture descriptions that are highly applicable in their later environment.	Customer	M
The responsibilities for realizing inter-operability are recorded before the development process is initiated.	Customer	M
When the development process does not result in architecture descriptions with high inter-operability, the architects responsible are held accountable.	Customer	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Repeatability</b>		
The development process is repeated consistently every time.	Maintainer, Governor	M
The repeatability of the development process results in consistent architecture descriptions.	Maintainer	M
The development process is well documented to facilitate consistent repeatability.	Maintainer, Governor	M
Constant improvement of the repeatability of the development process is pursued demonstrable.	Governor	M
The development process is sufficiently transparent to observe repeatability and its flexible enough to make improvements.	Governor	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Understandability</b>		
There are process description and/or experienced colleagues available to guide architects in gaining understanding about the development process and its logical concepts.	Executor	D
It does not require more than reasonable effort for architects to understand the development process and its logical concepts.	Executor	D

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Learnability</b>		
There are study materials and /or experienced colleague's available to aid architects in learning to use the development process.	Executor	D
It does not require more than reasonable effort for architects to learn to use the development process and its logical concepts.	Executor	D

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Operability</b>		
The executing architects are capable of executing the development process with a level of confidence.	Executor	M
The executing architects react purposeful and determined in case of the occurrence of errors and/or failures during the development process.	Executor	M
The development process is sufficiently clear and stable to maintain the trust of architects.	Executor	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Analyzability</b>		
The development process is sufficiently transparent to trace the cause of errors, failures and/or unexpected feedback within the development process.	Maintainer	M
Architects can trace the error, failures and/or unexpected feedback back to the source with no more than reasonable effort.	Maintainer	M
Executing architects are available to provide information about the development process.	Maintainer	D

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Modifiability</b>		
Errors and/or failures that are observed during the execution of the development process are addressed immediately.	Maintainer	M
Solving errors and/or failures within the development process does not require more than a reasonable effort from the architects.	Maintainer	M
As a result of immediately addressing errors and failures when they occur, the resulting architecture descriptions contain less errors and failures.	Maintainer	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Failure frequency</b>		
The development process includes procedures to record errors and/or failures during its execution.	Governor	M
Data about the number and regularity of errors and/or failure occurrences is used to improve the development process itself.	Governor	M
The responsibilities for improving the development process based on the feedback about errors and/or failures are recorded unambiguously.	Governor, Owner	M
Architects strive demonstrable to reach a low frequency of errors and failures of the development process.	Owner	M
The development process is constantly improved based on the feedback on errors and/or failures.	Owner	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Time behavior</b>		
There are guidelines in place about the quantity of time executing architects are allowed to use during the execution of the development process.	Owner	M
There are procedures in place to monitor and report the time usage by executing architects during development execution.	Owner	M
When the guidelines for time behavior are is not met, the architects responsible are held accountable.	Owner	M

<b>Statement</b>	<b>Viewpoint(s)</b>	<b>Status</b>
<b>Resource behavior</b>		
There are guidelines in place in the development process about the quantity of resources executing architects can use during process execution.	Owner	M
The development process includes procedures to monitor and use the usage of resources by executing architects.	Owner	M
When the guidelines for resource behavior are not met, the architects responsible are held accountable.	Owner	M