MASTER THESIS

QUANTIFYING THE CONTRIBUTION OF ENTERPRISE ARCHITECTURE TO BUSINESS GOALS

Alexandra Roditi

BUSINESS INFORMATION TECHNOLOGY
SCHOOL OF MANAGEMENT AND GOVERNANCE

EXAMINATION COMMITTEE
Dr. Maria-Eugenia Iacob
Prof. Roel J. Wieringa
Dr. Lianne Bodenstaff
MSc. (PhD Candidate) Wilco Engelsman

BiZZdesign

UNIVERSITY OF TWENTE.

20/10/2014

MASTER THESIS

QUANTIFYING THE CONTRIBUTION OF ENTERPRISE ARCHITECTURE TO BUSINESS GOALS

Author	**Alexandra Roditi**
Study	Business Information Technology School of Management and Governance
Student Number	1347772
e-mail	alexandra.roditi@gmail.com

Graduation Committee

First supervisor: Dr. Maria-Eugenia Iacob	University of Twente, *School of Management and Governance*
Second supervisor: Prof. Roel J. Wieringa	University of Twente, *Faculty of Electrical Engineering, Mathematics and Computer Science*
External supervisor: Dr. Lianne Bodenstaff	BiZZdesign
External supervisor: MSc. (PhD Candidate) Wilco Engelsman	BiZZdesign

Acknowledgements

Completing this thesis project was the last phase of my two-year journey for gaining my Master degree in the University of Twente. These past two years were full of new experiences which provided me with new knowledge and a new way of thinking and working. Working on this thesis helped me familiarize with fields that I had limited previous experience on and made me realize my interest in analysis and conceptual modeling. I believe that this thesis has helped me build on my professional skills and prepared me for my future career steps.

There are a number of people I would to like to thank for their contribution throughout this attempt. First of all, I would like to thank all my thesis supervisors Dr. Maria-Eugenia Iacob and Professor Roel Wieringa, from the University of Twente, and Wilco Engelsman and Lianne Bodenstaff, from BiZZdesign, for their guidance and support. Their critical questions and different perspectives on the topic pointed me towards the right direction, helped me understand what I wanted to achieve, assisted me in overcoming various obstacles and promoted the successful completion of this effort.

I would also like to express my gratitude to all my friends who made this journey a very pleasant experience and filled it with great memories. So, I would like to especially thank my friends and colleagues from the BIT programme, Iwan, Diana and Jurjen for the hard working times and for all the fun we had together during the last two years, as well as my colleagues from BiZZdesign who made the days at the office more cheerful and motivating. Of course, I owe special thanks to my friends who shared their evenings and weekends as well as many worth-remembering moments with me and made the Netherlands feel like home: Giorgos, Sofia, Afrodite and Katerina; and Ioanna for her magnificent friendship and understanding while experiencing this journey together and yet apart.

Above all I would like to thank my parents Haroula and Alkiviadis, and my brother Giorgos for their unconditional love, endless support and priceless advice which gave me strength throughout my whole journey, from the decision to come to the Netherlands till accomplishing my goals and completing my studies.

Finally, I would like to mention that studying in the Netherlands for gaining my Master degree was achieved due to the scholarship I received from the State Scholarships Foundation, and I quote:

"*The completion of this thesis project was co-financed by the Act "Scholarships Foundation Programme SSF (State Scholarships Foundation) for Postgraduate Studies (Master) – Horizontal Act", from resources of the Operational Programme "Education and Lifelong Learning", of the European Social Fund (ESF), the NSRF 2007-2013.*"

Alexandra Roditi

Enschede, October 2014

Management Summary

Organizational change can be imposed by a variety of reasons which entail uncertainty for the organization. To realize the change, transformations of the existing enterprise architecture are necessary. Since more than one alternative solutions may exist for addressing the change need, an analysis is essential for deciding which alternative will bring the largest possible benefits to the organization. Thus, the development of a business case is mandatory for evaluating the alternatives in terms of costs, benefits, resources and time. A business case, though, needs to be built on data and measurements derived from the enterprise architecture which can be analyzed and which can provide more educated and accurate decisions to the organization.

A change is motivated through business goals and, hence, the transformations in the architecture should contribute to the satisfaction of the goals. In other words, the expected impact of the transformation, that will take place in the architecture level, should be also visible in the goal level. With the help of the ArchiMate® modeling language, business goals can become tangible, since the systems, processes, business units or data objects that are associated with the realization of the goal can be traced. Thus, the effectiveness of a solution choice can be mapped to the related goals and the solution's contribution can be quantified.

For supporting the quantitative analysis in the goal domain, the EA-based Goal Quantification method is introduced. The method aims to combine a variety of techniques and analysis models in the goal level and the architecture level and to provide guidelines to the organizations for performing measurements and quantitative assessment based on indicators for identifying the most suitable solution. A performance indicator analysis through performance indicator structures represents the analysis in the goal level and guides the analysis in the architecture level as well. The analysis in the architecture domain is supported by architecture analysis models, which determine the objects of the enterprise architecture that are essential for the measurements. The focus in the architecture domain is on cost and performance aspects.

The method consists of ten steps, each one of which comprises a set of activities. The order of the steps is designed in such a way to facilitate the application of the method by the organization and to indicate a logical flow of the actions that need to take place for quantifying the change.

The proposed method is showcased through the ArchiSurance case study. While the focus of the original case is on costs, during the demonstration of the method, apart from costs, performance measures have also been selected to illustrate the capability of the method to support both cost analysis and performance analysis. Furthermore, an example algorithm in pseudocode is provided for demonstrating the feasibility of the method to be implemented in a tool.

Finally, the method is validated by means of interview sessions with five practitioners in the Enterprise Architecture field. The purpose of the interviews is to provide feedback with respect to the perceived ease of use, ease of understanding, usefulness and practical applicability of the method.

Table of Contents

Index of Figures

Index of Tables

Abbreviations

ABC	Activity-Based Costing
AF	Architecture Framework
AGORA	Attributed Goal-Oriented Requirements Analysis
ARR	Accounting Rate of Return
ATAM	Architecture Tradeoff Analysis Method
BIM	Business Intelligence Model
BRM	Business Requirements Management
BVA	Business Value Analysis
CBAM	Cost-Benefit Analysis Method
COCOMO	Constructive Cost Model
DepRVSim	Requirement Volatility Simulation considering Dependency relationships
EA	Enterprise Architecture
EAGQ	EA-based Goal Quantification
EAM	Enterprise Architecture Management
GORE	Goal-Oriented Requirements Engineering
GQ(I)M	Goal Question Indicator Measure
GQM	Goal Question Metric
IRR	Internal Rate of Return
IT	Information Technology
KPI	Key Performance Indicator
LCC	Life Cycle Cost(ing)
NPV	Net Present Value
OMSD	Optimum Measures Set Decision
PBS	Product Breakdown Structure
PI	Performance Indicator
PP	Payback Period
PPI	Process Performance Indicator
PrI	Profitability Index
QA	Quality Attribute
QOC	Questions, Options and Criteria
RE	Requirements Engineering
RM	Requirements Management
ROI	Return On Investment
SMO	Software Measurement Ontology
SPGQM	Structured Prioritized Goal Question Metrics
TCO	Total Cost of Ownership
TDABC	Time-driven Activity-based Costing
TSTS	Times Savings Times Salary
WBS	Work Breakdown Structure

CHAPTER 1

1. Introduction

This research document comprises the Final Master Thesis Project for the Master Business Information Technology programme which is conducted at BiZZdesign in Enschede. The focus of the research is to measure the Business Goals change through Enterprise Architecture by providing a unified analysis in both the Enterprise Architecture (EA) level and the goal and requirements specification level for assessing the value of Enterprise Architecture. In this chapter, the Research Goals and Research Questions that guide this research are defined and the description of the problem that this research aims to solve is also provided. Additionally, the Research Scope and Objectives are clarified and the research methodology which will be followed is described as well. Finally, an overview of the following chapters is given.

1.1. Research Goals

The main goals of this thesis are the following:

1. To provide one or more **mechanisms** in order to **quantify the links** between business goals and architectural elements in the **ArchiMate®** EA model, when a business change is in stake, by using the knowledge and information provided through traceability links
2. To identify **why** and **how** these mechanisms can improve the analysis of business goal changes
3. To prove the applicability of the designed mechanism(s) in the **Architect tool**.

1.2. Research Questions

In order to be able to achieve the above mentioned research goals, research questions need to be identified which will guide the exploration, design and development of the quantification mechanism(s) as well as the clarification of related topics and concepts. The main research questions and their sub-questions are the following:

RQ1. Why are quantifiable assessments of EA and Goal models needed?

- *Who are the stakeholders?*
- *What are their goals and differences regarding their understanding of EA?*
- *What is the problem(s) that needs to be addressed by quantifying the business goals change?*
- *What is the scope and context of this research?*
- *What are the purpose and the objectives of this research?*

The answer to this question will provide the background of the research and the motivation for continuing through the investigation of different techniques. By investigating the problem behind the research, the business value of the thesis outcome will be also established and the stakeholders to whom the new methodology will be more valuable will be identified. Additionally, establishing the scope of the research and its context will facilitate the exploration and provide direction in the following chapters. Thus, the contribution of this thesis will be addressed as a starting point for the research.

RQ2. What are the existing quantification techniques/models/mechanisms for expected cost and benefit for

a) Goal models?

b) Architecture models?

- *What are the analysis methods in each kind of model?*
- *What attributes of an element in each model can be measured?*
- *What are the quantification techniques for the analysis methods?*
- *How can the value of an element be calculated?*

By answering this question, a number of available techniques is expected to be found, both in scientific research and in practice. Since the question is twofold, the techniques, found in literature and practice, have to cover both the analysis on the EA level and on the goals and requirements level. Apart from the analysis techniques, the quantification mechanisms that relate to them or complement them should be found as well. The expected quantifications will address the costs and benefits which are related with elements in both models. Exploring the quantification techniques will assist us in identifying the attributes that can be measured in each model regarding costs and benefits, what is the methodology used for measuring these attributes, what is the relevance that these techniques have with EA and what is the relationship between the measurable attributes in the different levels of analysis. Additionally, an assessment is necessary in order to identify which of the quantifiable techniques, attributes and factors or indicators are relevant for this thesis, which leads to the next research question.

Since 'quantification' is a wide term and can include a wide range of topics, some limits should be put as well. Thus, fields, streams or theories relevant to Information Technology (IT), software engineering, Enterprise Architecture and ArchiMate®, change management, goal and requirements management and engineering, tradeoff and financial analysis will be mostly explored.

RQ3. Extend ARMOR with quantification mechanism(s).

- *Are there available techniques specific for the ArchiMate® model?*
- *Which of the identified techniques are useful for the ArchiMate® model?*
- *Which of the identified techniques can be adapted to EA and ArchiMate® motivation extension?*
- *How can the different analysis and quantification techniques of the two levels (EA and goal) be combined into quantification mechanism(s) of goal change?*

The third research question and its sub-questions aim at distilling from the identified techniques the ones that are relevant to EA and the ArchiMate® motivation extension and those which can be adapted to the elements of the ArchiMate® model in total. ARMOR is the language which describes the ArchiMate® motivation extension and which will serve as the basis for applying the quantification mechanism(s). Additionally, after selecting the techniques considered more applicable in ArchiMate®, both in the architecture model and also in the motivation extension, a combined mechanism is preferred to be proposed. Thus, exploration of available ways of combining these techniques and operationalizing them should be performed. In that way, the proposed mechanism will have a unified form and it will be easier to be supported by a tool. The implementation phase of the techniques is handled by the next research question.

RQ4. Design the implementation of the quantification mechanism(s) in the Architect tool.

Tool support is an essential aspect to be considered while designing the quantification mechanism in this thesis. One of the main goals of the research is to be able to apply and operationalize the new mechanism(s) in the Architect tool.

RQ5. What is the contribution of the quantification mechanism(s)?

- *Can the quantification mechanism(s) be used for making a prediction?*
- *For which kind of companies are the mechanisms useful?*
- *What is the contribution of the mechanism(s) in terms of the goals of this research?*

The quantification mechanism(s) designed in the thesis will have to be evaluated as well. So, the validation phase of this research is an important step for clarifying whether the initial goals have been addressed and what is the degree of contribution of the designed mechanism(s). The minimal requirement would be to enable prediction of the future value of a choice based on a business change. Another important issue is the identification of the types of companies or industries for which the quantification mechanism(s) will be applicable. The contribution and the business value of the thesis outcome can also be demonstrated through this assessment.

1.3. Problem Description

The research topic and consequently the conduct of the research are based on an observation and a business need proposed by BiZZdesign. BiZZdesign offers Business Requirements Management to its customers, which provides the refinement of business goals and stakeholder concerns into requirements, and combines it with Enterprise Architecture models, in which the link between requirements and the model can be defined. While these linkages are capable of supporting the determination of the impact of business goal changes on and through EA models, this capability is not utilized. On the contrary, currently, an architect uses the traceability links as a means to identify the elements that relate to the change by just following the links.

Thus, there is a need for determining the impact of change by using measures based on the traceability links which are already available. Before engaging in further discussion about how to derive and develop mechanisms for measuring the impact of change on and through EA, the reasons behind the need to quantify this impact should be described. The quantification, consequently, covers both the EA level and the goal level, which represents the motivation originating from the business or the organization. So, the need for assigning values to business goals will be examined in addition to the need of assigning values to EA models.

As it is mentioned by Iacob et al. [1], the EA discipline has emerged in order to address the need of maximizing the business value from IT investments. Thus, EA tries to bridge the gap that exists between business and IT by providing an improved overview and an explicit identification of the relationships among architectures regarding business processes, applications, information and technical infrastructure [2]. In the literature, a variety of benefits have been attributed to EA besides its ability to bridge business and IT. Some of the most common ones can be categorized under the following groups: (i) increased responsiveness and guidance to change, (ii) improved decision-making, (iii) improved communication and collaboration, and (iv) reduced costs [3]. In addition, Engelsman and Wieringa [4] mention that EA has been used for coordinating IT projects and managing IT costs, while an increase is observed in EA's contribution to improve the flexibility of the organization and to provide reasoning of how IT can assist in business goals realization.

Building an EA model, though, is not an easy task, since organizational change and significant resources are necessary for accomplishing it [1]. The demonstration of the value of the EA is another issue that has attracted attention and which is admitted to be a very complex and difficult process for organizations [5]. The economic nature of the organizations and the fact that EA is still viewed as an abstract concept by organizations and many stakeholders related to it, have increased the need to provide a concrete representation of the value of the EA [5]. The difficulty of measuring the impact, direct or indirect, of EA on different areas of an organization, the fact that, as a project, building an EA needs "time, money and effort to design, initiate, and embed it within the organization" [5], as well as the need for better understanding from the stakeholders in order to support and approve EA projects, necessitate the development and identification of clear measurements that will be able to assess the value of EA.

According to van der Raadt et al. [6], the EA function includes all the organizational activities as well as the stakeholders, in terms of roles and departments, that are involved in the EA decision-making process at enterprise, domain, project and operational levels. As EA decision-making activity, the authors consider the approval of changes on EA products, when either the change concerns a new product or a changed one, and the mechanisms that support the enforcement of the EA decisions on the organizational level. The stakeholders that either participate in the decision-making activity or are affected by the decisions are categorized in Table 1 according to their aspect areas, i.e. business, information, information systems and technical infrastructure, and organizational levels (enterprise, domain, project and operational).

In the context of this thesis, there are two stakeholder categories which are distinguished based on the stakeholders' interests and understanding on EA. Firstly, the stakeholders from the aspect areas of information, information systems and technical infrastructure which are at the domain, project and operational levels will be considered as the *IT management*. IT management has more understanding on the concept of EA because of their more detailed knowledge on and closer involvement in aspects or topics regarding IT systems, software applications and infrastructure components; the way these systems support different domains; the components' operational performance; the requirements related to the components and all the necessary activities that undergo their maintenance and improvement [6]. Secondly, stakeholders categorized in the business aspect and at the enterprise level comprise the *Boardroom* category. The people in this category are more concerned with strategic decisions regarding the future EA, but they have limited understanding and interest in the actual EA model. Even if EA models are considered as high-level representations of the architecture landscape in an organization, they can be quite detailed and not valuable from these stakeholders' perspective. A direct translation of EA models, in terms that these stakeholders are more familiar with (e.g. costs, risks, values), is expected to promote the perceived usefulness and value of the models in the boardroom and enhance the decision making process by enabling them to interpret different investment options and make comparisons among them.

	Business	Information	Information Systems	Technical Infrastructure
Enterprise	• CEO, CFO, COO	• CIO	• CIO	• CTO
Domain	• Head of Business Division (BD) or Business Unit (BU) • Business change manager	• DIO – Division Information Officer • IT change manager	• DIO • IT change manager	• Platform manager • Platform subject matter expert
Project	• Business project Manager • Business process designer	• Information analyst	• Software development project manager • Software designer/architect	• Infrastructure project manager • Infrastructure engineer
Operational	• Operational business manager • Business process engineer	• Data administrator	• Application management • Application administrator	• Data center management • Infrastructure administrator
				** IT management* **and** *Boardroom*

Table 1 - Key EA Stakeholders, their aspect areas and organizational levels[1]

EA models have been developed in order to support the representation of the relationships among the three different architectures, namely business processes, application and technology. ArchiMate® is an existing standard for enterprise modeling which allows modeling of the products and their services as well as the "processes, applications and technology that implement the services" [7]. According to Iacob and Jonkers [2], the focus of these models is mostly on functional aspects. The ArchiMate® modeling framework has been recently extended with a fourth aspect, namely the motivation aspect [7]. Through that aspect, traceability of business goals to IT architecture can be supported, since it facilitates tracking and justifying the influence of abstract (business) goals on more concrete IT-related goals and in turn on architecture components, and vice versa [4] [7]. The motivation extension of ArchiMate® has been realized by ARMOR, which is a requirements language with a notation and tool support for creating integrated goal models and EA models [4]. Additionally, BiZZdesign Architect is an EA tool that can support the ARMOR requirements language and the analysis of impact change through traceability links [8]. Enabling the modeling and visualization of the relationships among motivations and concerns of stakeholders, their goals and the architecture elements that realize these goals is very crucial for promoting new ways of analysis from the requirements domain [7].

Iacob and Jonkers [2] have pointed out that while quantitative properties of EA models are important in the EA level, they have not been studied sufficiently. They also state that "the availability of global performance and cost estimates in the early architectural design stages can provide invaluable support for system design decisions, and prevent the need for

[1] Source: van der Raadt et al. [6], p.22

expensive redesigns at later stages" [2]. The availability of such measures in the beginning of an EA project, which is usually triggered by a business need change, can also enhance the commitment to the project by the relevant stakeholders as well as provide sufficient evidence for investing in the project [5]. Thus, except from financial metrics, operational metrics and performance indicators [5] can contribute to the need of providing more tangible and quantifiable assessments of EA.

ArchiMate® modeling language provides the relationship between the EA model, which covers the business, application and infrastructure layers, i.e. the more tangible architecture layers, and the business goals and requirements which drive the design decisions. Except from the need for assessing the value of EA, which is necessary, transferring this value in the higher level which consists of the business goals and requirements is also essential. While assuming that EA models are less relevant for the stakeholders in the boardroom, it can also be assumed that the business goals are more relevant and valuable for the same people, as they engage more in defining these goals, deciding their prioritization as well as giving the 'green light' for their realization. Since EA projects are complex projects which can lead not only in technology improvements, but also in organizational changes, they can be categorized as 'Technochange situations', as defined by Markus [9]. In such situations, the expected outcome is an improvement of the organizational performance. Moreover, in 'Technochange situations', factors related to non-technology changes and to higher level objectives, which will contribute in gaining the expected benefits, are central during the consideration of alternatives. Since, change management can involve conflicts and resistance from different stakeholders, being able to assign values related, for example, to costs, benefits, resources, completion or transition time, directly to business goals can be proven beneficial and rather useful during the decision-making process.

When high-level goals are associated with specific values, their ambiguity level decreases, more clarity regarding the effort to accomplish them is achieved, the comparison among different approaches or scenarios realizing a goal is facilitated and the prioritization of a number of goals involved in a project or change can be assisted as well. Additionally, according to Karlsson and Ryan [10], priorities are valuable, since they enable the focus on the development process, more efficient project management, making more acceptable tradeoffs among conflicting goals and allocating resources based on the importance as represented by the prioritization. Furthermore, goal modeling contributes in "heuristic, qualitative or formal reasoning" [11] during the requirements engineering process which is used, among other purposes, for refining high-level goals to lower-level requirements. Thus, analyzing goal models and assigning values to goals by using the goal-graph representations can add a quantitative perspective to the reasoning supported by them.

Another important driver for this thesis is the lack of relationship between the existing analysis techniques in the EA modeling level and in the goal modeling level. While there is a relationship between the two levels through the ArchiMate® motivation extension, the analysis in each level and the conclusions or decisions made for each level are independent. The same stands for the quantifications in each level of analysis, regarding for example costs, risks, benefits. Consequently, a mapping between the two types of modeling and analysis would be a step towards unifying the analysis methodology across the entire EA model.

An overview of the pains and issues identified from the side of the research community and the organizations which adopt and use EA as well as the expected gains from incorporating quantifications and value assessment of EA are presented in Table 2.

To conclude, EA has been enriched by the introduction of the ArchiMate® motivation extension which can facilitate tracing the impact of the stakeholders and business goals to the architecture elements that realize them. By taking advantage of the ArchiMate® modeling framework, its ability to support impact analysis and the Architect tool that contributes to that task, an attempt to provide more advanced and concrete mechanisms for quantifying the value of EA and specifically measuring the impact of business goal change through consequent EA model changes is presented in this research. Providing measurements and quantifiable assessment of a business change is expected to enable the improvement of managing IT projects in relation to EA and the justification of the effect of a business change on EA in a tangible manner. Furthermore, quantifying the impact of change on EA models would assist in improving the analysis, the selection of design alternatives, performing a better tradeoff analysis and providing a more concrete rationale of the design decisions.

<table>
<tr><th>Pains</th><th>Expected Gains</th></tr>
<tr><td colspan="2">Research Community</td></tr>
<tr><td>1. Limited studies in quantitative attributes of EA</td><td rowspan="2">✓ Support the research on the field of EA and Goal analysis
✓ Provide a technique for relating quantification of the EA and Goal levels from both design and analysis perspectives</td></tr>
<tr><td>2. Lack of relationship/mapping between existing analysis techniques at:
• EA level
• Goal modeling level</td></tr>
<tr><td colspan="2">Organizations</td></tr>
<tr><td>1. Difficulties in building EA models with respect to:
• Assessment of organizational change
• Resources utilization and distribution</td><td rowspan="3">1. Availability of global performance, benefit and cost estimates;
and
2. Tangible and quantifiable assessments of EA and Goals
can:
✓ Support EA design decisions
➢ analysis and selection of design alternatives
➢ tradeoff analysis
➢ more concrete rationale
➢ easier comparison among different approaches or scenarios realizing a goal
➢ justification of the effect of a business change on EA in a tangible manner
✓ Support Goal prioritization
➢ through assigning values to goals
✓ Support for quantitative reasoning of goal decisions
➢ decrease of goal ambiguity level
✓ Provide sufficient evidence for investing in (EA) projects
➢ improve interpretations on and assessments of investment options
✓ Improve EA project management
➢ allocating resources based on goal prioritization (values)
✓ Promote the perceived usefulness and value of EA and goal models in the boardroom</td></tr>
<tr><td>2. Difficulties in demonstrating the value of EA to high-level management with respect to:
• goal assessment
• organizational change
• decision-making</td></tr>
<tr><td>3. Difficulties in measuring the impact of EA on the organization
• Decision-making for an EA project
• Assessment of expected benefits value
• Assessment of goals/drivers value
• Assessment of EA project implementation (costs, time, effort)</td></tr>
</table>

Table 2 - Pains and Expected Gains from Quantifications in Goal and EA modeling

1.4. Scope, Context and Objectives of the Research

The scope of this research as well as its objectives will be clearly defined in this section. Defining them will serve as a guideline through the literature research and the design decisions in the following chapters.

The scope of this research is to identify and define quantification techniques which will estimate and calculate the expected costs and benefits of EA changes as imposed by changes in business goals. Since business goals are the drivers for design and implementation decisions, the expected outcomes of these decisions are anticipated to be interpreted in quantifiable terms in both levels of analysis, EA and Goal modeling. Additionally, the described impact analysis might need to be adjusted in order to be applied in model representations. 'Benefit' is an umbrella term which can be decomposed in a number of different aspects, attributes and measures. The determination of the term 'benefit' has to be specified in the context of the analysis to be performed. Furthermore, since this research is conducted at BiZZdesign, the incorporation of the quantification techniques in ArchiMate® context and realization of them in the Architect tool are also pre-requisites.

The context that the techniques are going to be used in is also of high importance for the identification of the most relevant ones. Information processing organizations comprise the focus area of this thesis, regarding the industry type that the mechanism(s) will be applicable on. This type of organizations can benefit from a quantification mechanism in gaining provision for improved management of the information flow in the organization. Additionally, the type of projects, that the mechanism will be used for, is important to be defined. According to Crundwell's classification of projects [12], different kinds of projects can be defined depending on the decisions that need to be taken regarding the project and the information that are important for the project. The mentioned classifications are the following:

- *Project interdependency*: whether the project in stake is related to other projects of the organization, e.g. a project may neutralize the realization of another.
- *Depth of evaluation*: the degree of the analysis required for making decisions on the project, e.g. very detailed.
- *Level of project risk*: depending on the purpose of the project (ranging from equipment replacement to the development of a new process), different risk levels can be identified, i.e. low, normal or high risk.
- *Mandatory or Discretionary projects:*
    - Mandatory projects can entail:
        - actions for maintaining service quality or process reliability
        - replacing equipment
    - Discretionary projects can entail:
        - creation of new business lines, processes, services, etc.
        - cutting costs

From the above categorization, it can be derived that quantification mechanisms can be more beneficial for projects that may involve dependencies, since the required analysis in these cases is more extensive and precise measures can enhance the trade-off analysis as well as portfolio management when a lot of projects are under consideration. Furthermore, when the risk of a project is more than moderate, a more detailed analysis supported by quantifications can help in lowering the risk of wrong decision-making. As for the last classification, both mandatory and discretionary projects can be benefitted, since a variety of aspects can be addressed, e.g. quality trade-offs, process development or replacement, cost reduction, profit maximization, etc. Thus, the context of the thesis includes both mandatory and discretionary projects that:

- may have relations with other on-going projects
- are of at least moderate risk
- need a more detailed analysis and not only a superficial evaluation.

Additionally, according to the distinction proposed by Peppard et al. [13], this thesis will address and support problem-based IT investments. In the problem-oriented interventions, there is a specific improvement target, which is the basis of the analysis and the estimations for the future situation. Based on the target, various ways and alternatives are examined in order to reach the goal by choosing the best option. The targets are derived through identified problems in the working process which will be removed by introducing new ways to perform the business processes and activities. This type of investments which aim to more exact results will provide a better input for the quantifications.

Finally, the objectives of the thesis represent the purpose of conducting the research and what must be achieved by the end of it. The following are the identified objectives, as derived from the research goals:

- ✓ Identify the existing analysis and quantification mechanisms, techniques, theories or models with respect to EA and goal modeling;
- ✓ Adapt or adjust techniques that are not specific for EA analysis (if necessary)
- ✓ Introduce one or more quantification mechanisms which will combine and operationalize these techniques;
- ✓ Provide an algorithm (pseudocode) which will analyze/demonstrate the quantification mechanism(s);
- ✓ Provide cost and benefit estimations for the affected elements in the EA and goal models;
- ✓ Demonstrate the implementability of the new quantification mechanism(s) in the Architect tool;
  and
- ✓ Verify that the new quantification mechanism(s) assists the analysis of business goal changes through analyzing EA models.

1.5. Research Approach

1.5.1. Engineering Cycle Theory

In this research project, the research methodology introduced by Wieringa [14] and revised by Wieringa and Morali [15] is adopted. This methodology comprises the regulative or engineering cycle which represents the notion of rational reconstruction of a practical problem solving process. This process is actually a chain of tasks from problem to improvement implementation that practitioners or researchers construct to justify the improvement to others [14]. Wieringa [14] defines IS design-science research as a nested problem structure, which consists of two problem classes, namely the practical problems and the knowledge questions.

A *practical problem* or *improvement problem* is defined "as a difference between the actual state of the world and the world as desired by some stakeholder" [15]. Thus, solutions to practical problems will impose changes in the current state of the world in order to satisfy the goals of the stakeholders and the process of finding the solutions includes investigating the goals of stakeholders and evaluating the possible solutions by applying stakeholder criteria [14].

Except from practical problems, Wieringa [14] defines *knowledge problems* and the relationship between knowledge and practical problems. Knowledge problem is defined as "a difference between current knowledge of stakeholders about the world and what they would like to know" [14]. Knowledge problems ask for information about something that exists, so they do not aim to change the world but only improve the knowledge about the world. Answering knowledge questions involves "applying knowledge from the knowledge base or by doing original research, using conceptual analysis or empirical methods such as experiments, case studies, field research or modeling and simulation" [14]. In contrast, solving practical problems entails "matching problems and solutions in a regulative cycle" [14]. According to Wieringa [14], a practical problem can be decomposed in knowledge sub-problems and practical sub-problems. An illustration of that is given in Figure 1.

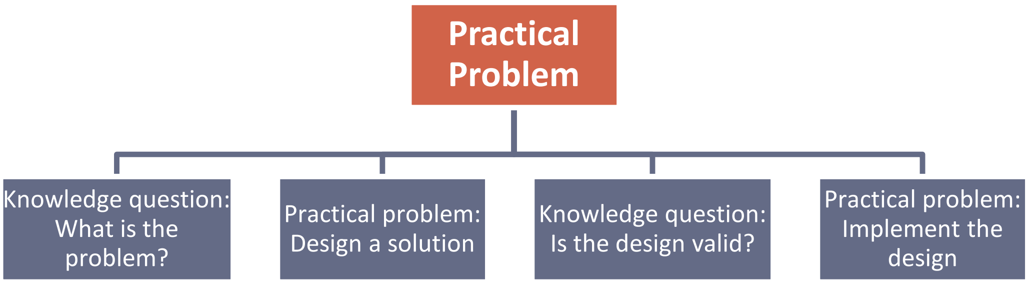


Figure 1 - A simple decomposition of practical problems into sub-problems, based on Wieringa [14]

The structure of the engineering cycle needs further explanation since it is the basis of this research. The engineering cycle starts with the investigation of a practical problem, which may be the outcome of previous practical problems, then solution designs are specified and validated and a selected design is implemented. The result of the implementation can then be evaluated, which may be the trigger for a new engineering cycle. Knowledge questions can be found in the problem investigation phase, due to the need of identifying existing problems and establishing an understanding of the problem; and in the design validation phase, where prediction of the effects of an implementation are examined [14]. In more detail, the four phases of the engineering cycle are the following:

1. **Problem Investigation.** In this phase an investigation for finding information and understanding the given problem takes place. The main objective of this phase is "to describe the problem, to explain it, and possibly to predict what would happen if nothing is done about it" [14]. So, in more detail, what needs to be defined are [15]:
   a. Identification of stakeholders and their goals, which are then operationalized into criteria.

   b. Investigation of relevant phenomena for the problem in stake, and
   c. Assessment of how well these phenomena agree with the goals of the stakeholders.
2. **Solution or Treatment Design.** After investigating the problem, one or more treatments are designed, which actually consist of the interaction between an artifact and the problem context [15] and they are considered as attempts to solve the problem of the first phase. The treatments are viewed as solutions, even if they are not specified completely. So, in order to reach the *end*, which consists of one or more stakeholder goals, the *means* to reach this end, i.e. the solution, should be designed [14]. In this research the solution will be a new or improved quantification mechanism, while the problem context in the case of this research is the organizational context and the changes in business needs and goals that it imposes which in turn affect the EA models. In Figure 2, the points that a researcher should consider in this phase are mentioned.
3. **Design Validation.** In this phase of the regulative cycle, the contribution of the specified design to the goals, that were set at the beginning, is examined [14]. Before implementing the design, validation of the design should be performed [15]. This phase consists of knowledge questions, which are of the four following types [14] [15]:
   a. *Causal question* or *Expected effects*: “What will be the effects of the artifact in a problem context?”
   b. *Value question* or *Expected value*: “How well will these effects satisfy the criteria?”
   c. *Trade-offs*: “How does this treatment perform compared to other possible treatments?”
      This question is applicable when more than one version of the treatment has to be compared or even when more than one treatment has been designed.
   d. *Sensitivity* or *External validity*: “Would the treatment still be effective and useful if the problem changes?”
      This part of the validation aims to assess the possibility of the problem becoming broader and larger.
4. **Treatment implementation and evaluation.** The final phase of the regulative cycle for one iteration, since the outcome of this part may initiate another round of steps, entails transferring the designed solution to the environment [15]. When the transformation has been achieved, then the evaluation can be performed by asking the same questions with the Design Validation phase, but this time the answers concern the actual effects in the real-world.

In Figure 2, the engineering cycle is illustrated. The first three steps of the cycle comprise the Design Cycle, which actually skips the ‘Treatment Implementation and Evaluation’ phase. When the validation of the designed treatment is completed, a new cycle is initiated.

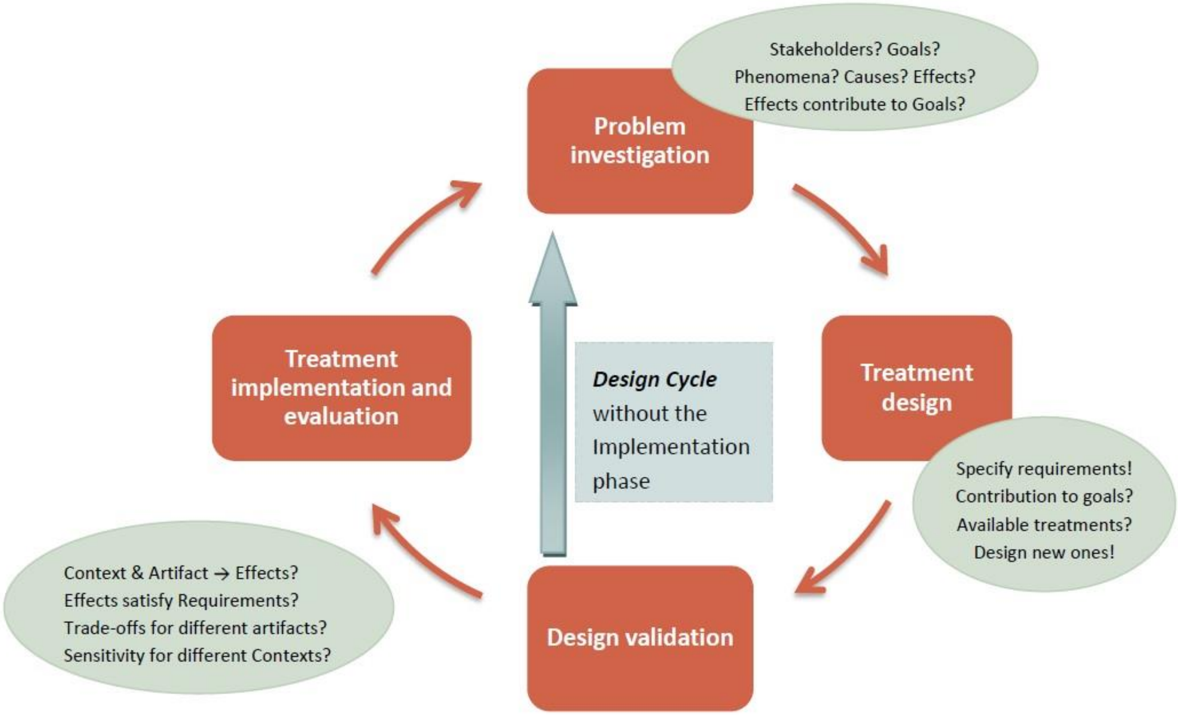


Figure 2 - Engineering Cycle and Design Cycle

1.5.2. Applying the Design Cycle

Based on the definitions of practical problems and knowledge questions, this research can be classified in IS design-science research; thus, the nested problem structure can be applied. The current research is a practical problem, since its goal is to develop new or improved techniques to support quantification of change, but knowledge questions are also deployed in order to facilitate the investigation of the current situation in EA and impact analysis, and the validation of the designed solution against the initial goals and concerns. By using the concepts of practical and knowledge problems, the objectives identified in Section 1.4. can be restructured and classified in these two categories, as it is shown in Table 3. It can be observed that most of the practical problems are a decomposition of the main problem which is the development of the quantification mechanism(s).

Practical Problems	Knowledge Questions
• Introduce one or more quantification mechanisms which will combine and operationalize existing techniques • Adapt the existing techniques in EA and goal modeling • Provide an algorithm (pseudocode) which will analyze/demonstrate the quantification mechanism(s) • Provide cost and benefit estimations for the affected elements in the EA and goal models • Enable the implementation of the new quantification mechanism(s) through Architect	• Identify the existing analysis and quantification mechanisms, techniques, theories or models in EA and goal modeling • Verify that the new quantification mechanism(s) assists the impact analysis of business goal change in EA

Table 3 - Practical Problems and Knowledge Questions of this Research

This research will follow the Design Cycle as a guideline for progressing throughout the development of the mechanism(s) (Table 4). The *'Problem investigation'* phase has been covered in the Problem Description section as well as in Section 1.4, where the stakeholders of the outcome of the thesis, the background of the problem, the scope and the drivers for this research have been established.

In the *'Treatment Design'* phase, the new mechanism(s) will be designed. This phase includes the investigation of available treatments. As a first step, a literature review will take place for acquiring the background knowledge for the related concepts to the research, e.g. the terms 'goals', requirements traceability and EA. Secondly, identifying the existing theories and techniques regarding analysis and quantification in both the EA and Goal level is the most important step towards the demonstration of the available treatments. Then, the development of the quantification mechanism(s) will be based on these findings. The design will rely on the contribution of the quantification mechanism(s) to the goal of measuring the impact of business change through EA models from both cost and benefit perspectives. A further investigation of the existing techniques, their tool support and their operationalization belongs in this phase as well, because it will serve as the basis for developing the new mechanism(s).

In the *'Design validation'* phase, which is the third and final one, the design methodology will be validated. The validation involves using a realistic problem as a test case. Ideally, the case should refer to a project that entails at least moderate risk and complexity and which would necessitate quantitative analysis in order to support the decision-making process. Such a situation would help proving the contribution of the mechanism(s) in similar situations. In the validation, expectations and opinions from professionals and experts will be gathered based on the outcome of the mechanism. An assessment of the contribution of the mechanism will take place as well. As the design science theory imposes, multiple iterations, especially in the second phase, are expected.

A final remark regarding the current research is that it may initiate a new Design Cycle. In the context of this thesis, the investigated problem is considered as a practical sub-problem of a larger Design or Engineering Cycle for enhancing and extending ArchiMate®. The aim of the new mechanism(s) is to be incorporated in the extension of ArchiMate®, which is constantly being improved, and to be incorporated in the Architect tool. A variety of viewpoints are being extended in the modeling language, so multiple parallel design cycles are in progress. One of them is the focus of this research.

<table>
<tr><th>Design Cycle Phase</th><th colspan="2">Research Questions</th><th>Thesis Chapters/Sections</th></tr>
<tr><td>Problem investigation</td><td colspan="2">RQ1</td><td>• Section 1.3: Problem Description
• Section 1.4: Scope, Context and Objectives of the Research</td></tr>
<tr><td rowspan="3">Treatment Design</td><td>Available treatments</td><td>RQ2</td><td>• Chapter 2*: Literature Review on Basic Concepts
• Chapter 3: EA Model Analysis Techniques
• Chapter 4: Goal Model Analysis Techniques</td></tr>
<tr><td rowspan="2">Design Questions</td><td>RQ3</td><td>• Chapter 5: An EA-based Goal Quantification Method</td></tr>
<tr><td>RQ4</td><td>• Chapter 6: Demonstration of the EAGQ Method</td></tr>
<tr><td>Design validation</td><td colspan="2">RQ5</td><td>• Chapter 6: Demonstration of the EAGQ Method
• Chapter 7: Validation of the EAGQ Method
• Chapter 8: Contribution, Recommendations, Limitations & Future work</td></tr>
<tr><td colspan="4">*Chapter 2 does not answer RQ2. It can be considered as part of the background analysis.</td></tr>
</table>

Table 4 - Mapping Overview: Design Cycle, RQs and Chapters

1.6. Structure of the Thesis

This research document consists of eight chapters:

1. **Chapter 1 - Introduction**
   a. Research Goals, Objectives and Research Questions
   b. Problem Description & Scope and Context of the Research
   c. Research Approach
2. **Chapter 2** - **Literature Review on Basic Concepts**
   a. Business Goals and Requirements
   b. Business Requirements Management & Requirements Engineering
   c. Requirements Traceability
   d. Enterprise Architecture: Frameworks, ArchiMate® model language and tool
3. **Chapter 3 - EA Model Analysis Techniques**
   a. Quantitative Analysis concepts
   b. Transformations between design and analysis space
   c. Benefit analysis
   d. Financial analysis techniques
   e. Cost analysis techniques
   f. Performance analysis techniques and Performance & Reliability measures
   g. EA-specific analysis techniques
4. **Chapter 4 - Goal Model Analysis Techniques**
   a. GQM and related methods
   b. Goal Analysis Frameworks
   c. Requirement change Impact analysis techniques

   d. Goals and Performance Indicators

5. **Chapter 5 - An EA-based Goal Quantification Method**
   a. Design of the proposed goal quantification model
   b. Description and analysis of each model step and its activities
   c. Incorporation of techniques presented in Chapters 3 & 4 in the model
6. **Chapter 6 - Demonstration of the EAGQ Method**
   a. Application of the method
   b. ArchiSurance Case Study
   c. Combination of cost and performance analysis
   d. Algorithm for implementing parts of the method
7. **Chapter 7 - Validation of the EQGQ Method**
   a. Evaluation method and Assessment criteria
   b. Validation results
   c. Personal reflection
8. **Chapter 8 - Contribution, Recommendations, Limitations & Future work**
   a. Contribution of the method: Answers to RQs
   b. Recommendations
   c. Limitations of the method and the research
   d. Future work

CHAPTER 2

2. Literature Review on Basic Concepts

According to the Design cycle, the first step is to investigate the problem context which has initiated the research and find information for understanding it. In this chapter, an overview of the related concepts is provided for establishing the background of the research and for presenting the definitions on which the research will be based.

2.1. Business Goals & Requirements

The main purpose of this research is to develop a methodology for quantifying changed or new business goals through Enterprise Architecture models. In order to be able to measure the value of goals, establishing what is meant by the term 'goal' is essential. The definitions, types, usefulness of goals as well as goal modeling languages are discussed in this section. Additionally, 'requirements' are used as the link between the goals and the Enterprise Architecture models, so identifying what a requirement is and which the relation between requirements and goals is, are mandatory.

2.1.1. Goal definition

In literature, goals have been defined from several perspectives regarding the business and the system under development. A few definitions are the following:

- Kavakli and Loucopoulos [16] define business goals in relation to business processes and they are presented as "business objectives aiming to create value to customers".
- "A goal is an objective the system under consideration should achieve." [11]
- ArchiMate® definition of goal: "some end that a stakeholder wants to achieve" [17]
- Other definitions of goals in the three main Goal modeling languages [18]:
    - KAOS: goals as desired system properties that have been expressed by some stakeholder(s).
    - i* and Tropos: goal as the intentions of a stakeholder.

The definition which is chosen in this research is the one proposed by ArchiMate®, since it combines a variety of definitions from the EA literature as well as the Goal-Oriented Requirements Engineering (GORE) literature [18] and it is the one clearly understood by architects when the design of a new model is under development [4]. Additionally, this is a more general definition which can incorporate the concepts of soft goal and hard goal, as the case study by Engelsman and Wieringa [4] has shown in practice. Finally, the definition of goal in the ArchiMate 2.1 Specification can represent high-level business goals as well as lower-level goals.

2.1.2. Goal Types

There are several types of goals found in literature. The most important and relevant ones for this research are the following:

- <u>Functional goals</u>: "Functional *goals* underlie services that the system is expected to deliver" [11]
- <u>Non-Functional goals</u>: "non-functional *goals* refer to expected system qualities such as security, safety, performance, usability, flexibility, customizability, interoperability, and so forth." [11]

- Hard goals: Two definitions have been found which complement one another, (i) "A hard goal is a goal with measurable indicators" [4], and (ii) "(hard) goals [are the goals] whose satisfaction can be established through verification techniques" [11].
- Soft goals: The two definitions found can serve as an explanation of each other, (i) "A soft goal is a goal without measurable indicators" [4], and (ii) "soft goals [are the goals] whose satisfaction cannot be established in a clear-cut sense" [11].

2.1.3. Requirements

Except from goals, the concept of requirements needs to be established as well. While a goal refers to the result that a stakeholder desires, a requirement is related to the system under development and it is defined as "a desired property that must be realized by a system" [8]. In the context of an existing problem, there is a relationship between the concepts of goals and requirements, as requirements are the translation of goals for a possible solution [8]. Furthermore, a requirement is presented as "some end that must be realized by a single component of the architecture" [4], while the realization of a goal may require the cooperation of multiple agents [11]. A requirement, though, still remains a goal with the difference that it is under the responsibility of a single agent or component. The case study by Engelsman and Wieringa [4] has also shown that requirements in practice are treated as a special case of goals and particularly as a goal assigned to an element. This observation can also be expected due to the definition of goals "as objective(s) the system under consideration should achieve" [11] mentioned above.

Since the ArchiMate® modeling language is going to be used in this thesis, the definition proposed by the Open Group for requirements is adopted in this research: "A requirement is (…) a statement of need that must be realized by a system" [17].

In the literature, definitions of functional and non-functional requirements are also found, which should be mentioned here in order to align them with the definitions of functional and non-functional goals. The definitions are the following:

- "Functional *requirements* define the functions or behavior that must be supported by some actor" [8]
- "Non-functional *requirements* define how well the functions or behavior must be performed by [an] actor" [8]

By presenting these definitions, it is obvious that a direct relation between the definitions of (non)functional goals and requirements exists.

2.1.4. Why are Goals necessary?

The usefulness and the reasons why goals are important have also been stated and discussed in literature. Goals are identified as an essential component in Requirements Engineering, as they can assist "traceability, conflict detection and resolution, and exploration of design choices" [19]. Moreover, goals are used as a criterion for completeness in requirements specifications as well as a criterion for pertinence [11]. Goals can also provide a better explanation to stakeholders and high-level goals can serve as more stable elements on which a design and a model can rely [11]. Furthermore, goals can be used for defining the responsibilities and actions of the agents or actors in a system [20].

2.1.5. Goal Models

The most cited Goal Models are i*, Tropos, and KAOS, which "represent an intentional ontology used at the early requirements analysis phase to explain the *why* of a software system" [21]. Goal modeling is considered an effective way of capturing stakeholders' expectations and the reasons that support the development of a system. "Goal models are intentional representations of users' goals and the ways users may adopt to satisfy them" and they can also represent quality attributes [21]. Goal models are techniques for materializing GORE. GORE is the approach of Requirements Engineering which supports analysis of the purpose of a proposed solution by demonstrating which goals realize or conflict with other goals and which goals facilitate the need that initiated the development of the solution [7].

The main characteristics of three popular Goal Modeling Languages are:

- **i***: The i* framework focuses on dependencies among actors in order to achieve a goal or a task, or to access a resource [11] [22]. Thus, the core concepts of i* model are the 'agent' and the 'goal'. Through the dependencies among actors, the i* framework can also represent the organizational context as the environment of the system under development as well as the internal relationships among the elements that represent the intentions of the agents relevant to the system [23].This model focuses on the early stages of requirements, where an understanding of the current organizational state needs to be established and an analysis of the need for change needs to be performed [24] [22]. The contribution of this modeling technique in early requirements analysis is that it enables capturing except from the *what* or *how* of developing a solution but also the *why* [22]. In that way, checks regarding, for example, the "viability of an agent's plan or the fulfillment of a commitment between agents" [11] can be supported and a more refined analysis of the dependencies in a system under development can be useful for treating requirements in a more uniform manner [22].
- **Tropos** is an agent-oriented software development methodology which covers the whole spectrum of the development process. It supports all the necessary analysis and design activities of this process. In Tropos methodology, five main development phases have been designed: *Early Requirements, Late Requirements, Architectural Design, Detailed Design* and *Implementation*. The goal of Tropos is building a model of the future system and its environment, which will be refined progressively and which will provide a common basis for the various activities during the development of the system. The main concepts of a Tropos model are: *actor, goal, plan, resource, dependency, capability,* and *belief*. One of the activities that take place in the *early requirements phase* is goal modeling, which emphasizes the analysis of actor's goals from the actor's point of view. In the *late requirement phase*, the focus is upon the future system within its operating environment, while the *architectural design phase* defines the agent-specific details regarding goals, beliefs, capabilities and interactions or dependencies. Thus, Tropos framework promotes an iterative refinement of actors and goals which is different in each phase.[22]
- **KAOS** is a formal reasoning approach which highlights the importance of explicitly representing and modeling organizational goals through refining high-level goals in a progressive manner [24]. This approach puts more emphasis on the later stages of Requirements Engineering and it assumes that there is sufficient knowledge about the

current organizational state, the change goals and the issues behind them in order to enable the formulation of the future solution which will address the issues [24]. In KAOS, a requirements model is built which consists of four sub-models: (i) the *goal model* which "captures the intentional view shared by all implied stakeholders", (ii) the *object model* which "captures the terminology needed to express the problem to solve", (iii) the *agent model* which "provides an agent-centered view on the system-to-be", and (iv) the *operation model* which "describes how agents have to cooperate to achieve the goals" [25].

2.1.6. Goal refinement process

There is a process through which goals are clarified and improved until they reach their final form before the development of the system. This process comprises a variety of steps which can be grouped in four main steps of Requirements Engineering (RE): (i) elicitation, (ii) analysis, (iii) specification, and (iv) validation [8]. According to Lamsweerde [11], the refinement process is decomposed in more sub-steps which can be mapped to the four main steps of RE. These steps include *goals identification, elaboration/elicitation*, which can be further decomposed in *refinement, abstraction* and *operationalization*, *verification, conflict management* and *negotiation*, and finally, *validation and alternative selection*. The mapping is not one-to-one and in the case of validation, as it is proposed by Lamsweerde [11], it can be mapped to both analysis and validation of the main RE steps, since it also involves the detection of inconsistencies which can be part of the completeness checks in the analysis step. In short, the four main steps of identifying and analyzing goals include the following procedures [8]:

- **Elicitation**: In this step, the identification of stakeholders, their concerns, the concerns' assessment, and the goals that address the assessments take place. The result is an initial requirements model.
- **Analysis**: The outcome of the elicitation phase is handed over to the analysis phase, where structuring and refinement in terms of consistency and completeness occurs. Resolution of conflicts, identification of dependencies, investigation of alternative solutions and prioritization of them are also responsibilities of this step. Moreover, the final model which is the outcome of this phase should be backwards and forwards traceable (see section 2.3).
- **Specification**: In this step, the actual representation of the models that have been produced in the two previous steps is performed. A modeling language is used for the representation.
- **Validation**: Quality assessment of the alternative solutions identified in the analysis step and selection of the best alternative are the goals of this step. The phases of elicitation and analysis may be revisited after the quality has been assessed. Different methods and indicators of quality may be used.

2.2. Business Requirements Management and Requirements Engineering

Business Requirements Management (BRM) plays a central role in the initial proposition made by BiZZdesign which functions as a starting point for this research. Better understanding of the term, what it contains and how RE, as a part of BRM, is operationalized, are described in this section.

2.2.1. Business Requirements Management and Requirements Management

The concepts of Requirements Management (RM) and BRM are defined in this section and the relation of BRM with RE is also clarified, since requirements are the main link between the business goals and the enterprise architecture elements. The definitions are the following:

- RM definition: "RM is the systems engineering activity principally concerned with finding, organizing, documenting and tracking requirements for systems. Its focus is maintaining traceability, defined as the 'ability to describe and follow the life of a requirement'" [8].
- BRM definition: "BRM denotes the early phase of the RM process that is concerned with the identification, description, analysis and validation of requirements at business level and their realization in enterprise Architecture" [8].

2.2.2. Requirements Engineering and RE Cycle

A very often stated term, which is also closely related and actually a subset of BRM, is RE. The definition of RE as well as a short reference to its sub-streams of Problem-oriented and Solution-oriented RE are presented below:

- RE definition: RE includes the activities of finding, organizing and documenting requirements [8]. Moreover, RE is positioned in the development phase of a system where requirements of different stakeholders are gathered and processed in order to be formally specified in a requirements specification [26]. There are two viewpoints regarding RE process:
    - *Problem-oriented RE*. Since RE is considered as a process of finding a solution to a problem, problem-oriented RE focuses on understanding this problem. So, emphasis is put on modeling and analysis of the problem domain, which also covers the investigation and documentation of it. [8] [7]
    - *Solution-oriented RE*. The other viewpoint is more related to solution specification [8]. From the solution-oriented RE perspective, "a requirements model describes the context of the system to-be, the desired system functions, their quality attributes, and alternative configurations or refinements of these functions and attributes" [7]. An analysis on the alternatives is also performed in order to identify the best solution to the problem [8] [7].

The identification of the two viewpoints of RE has also led to the notion of the *Problem Chain* [8]. Since problem-oriented RE and solution-oriented RE are considered complementary, they can be used in iterations for refining a problem progressively. When a problem is linked to a solution, the solution can be treated as a new problem for the next level of analysis. In that way, a problem chain is created. The notion of Problem Chain is closely related to linking business goals to Enterprise Architecture (EA) elements, which eventually realize the goals. The business goal side can be considered part of the problem-oriented RE viewpoint, while the realization by an EA element can be attributed to solution-oriented RE.

According to Engelsman et al. [8], RE can be decomposed in three steps (Figure 3):

1. **Problem investigation:** This steps aims to identify all relevant stakeholders, their concerns and assessments of the concerns and provide an elicitation, decomposition and initial structure of high-level goals of the problem, which can be an organizational change, for example. A series of sub-steps and iterations are defined in order to achieve the desired set of stakeholders, concerns, assessments and goals.
2. **Investigate solution alternatives**: In this step, emphasis is put in finding solutions by using as an input the goal models and structures designed in the previous steps. This step of the cycle focuses on refinement of goals into requirements, identification of conflicts and finding solution alternatives. The most important outcomes of this step, especially for this research, are the prioritization of goals and requirements as well as the relation to the architectural elements that realize the requirements.
3. **Solution Validation**: The final step of the RE Cycle includes the investigation of the alternatives identified in the previous step and the decision of the best solution. A possible outcome of this step could trigger new iterations of the ER Cycle.

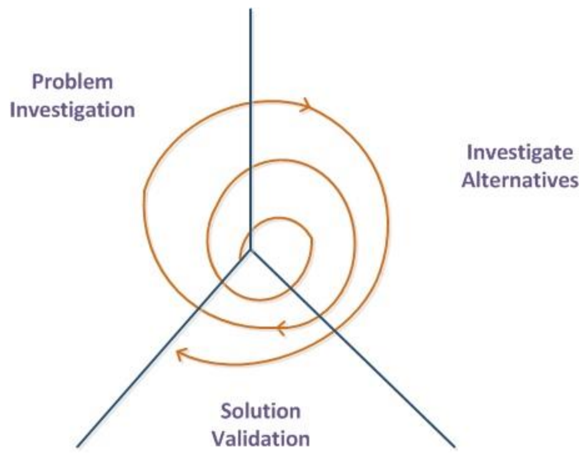


Figure 3 - RE Cycle

2.3. Requirements Traceability (RT)

This research and the outcome of this research rely on the current ability of architects to follow traceability links between goals and artifacts. As it is mentioned by Quartel et al. [7], in the advent of a change, traceability is important for EA for supporting the analysis of the impact of the change by following the relation between abstract goals and more concrete goals, between design artifacts and abstract goals and vice versa. Thus, further understanding in RT is needed by defining traceability and its aspects.

2.3.1. RT Definitions

- "Requirements Traceability refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction, i.e. from its origins, through its development and specification, to its subsequent deployment and use, through periods of on-going refinement and iteration in any of these phases." [8]

- Traceability is "the means whereby software producers can 'prove' to their client that: the requirements have been understood; the product will fully comply with the requirements; and the product does not exhibit any unnecessary feature or functionality." [26]

Backwards RT: "Backwards traceability is the ability to trace a requirement back to its source, e.g., a stakeholder, concern or even another less detailed requirement, which motivates or justifies the requirement." [8]

Forwards RT:

1. "Forwards traceability is the ability to trace a requirement to the components of a design or implementation that realize the requirement, for example more detailed requirements, processes, services and IT systems." [8]
2. "Forward traceability, in which artifacts (...) constrained by the requirements specification need to be traced back to the requirements specification." [26]

2.4. Enterprise Architecture Frameworks, Models and Tools

In this section the concept of Enterprise Architecture will be discussed in order to establish a firm background regarding the essential elements of it, the frameworks around it, the models that have been developed to support it and the tools that are currently used.

2.4.1. Enterprise Architecture

In literature, architecture is described as a means for organizing the selection of components and architectural elements of a system, their interaction and constraints, and the principles that govern the design and evolution of the system, in order to deal with uncertainty and change by bridging the perceptions and concerns of the stakeholders and the possible practical solution [8] [7] [27] [28]. Additionally, architectures for an enterprise can also facilitate the insight and overview required to successfully align business and IT [28]. Thus, bridging the gap between business and IT necessitates the design and maintenance of Enterprise Architecture (EA), which can be defined as:

"Enterprise Architecture (EA) is the complete, consistent and coherent set of methods, rules, models and tools which will guide the (re)design, migration, implementation and governance of business processes, organizational structures, information systems and the technical infrastructure of an organization according to a vision." [8]

Since EA provides the elements that comprise the system, it can be seen as the blueprint from which the system can arise and as a 'skeleton' of the "essential features and characteristics of the system" [29]. Except from uncertainty, EA can also assist in managing complexity and risk which may appear due to the involvement of a variety of factors, e.g. technology, size, interface, context, stakeholders [29].

From the definition of EA, it can be realized that EA comprises a wide range of elements which can be grouped in three types of architectures, which are also the layers of EA. Based on the description provided by Engelsman and Wieringa [4]: (a) an architecture of the business can include products, services and processes, (b) an application architecture can include application components, functions and services, and (c) an infrastructure architecture can include servers, mainframes, network. These layers are built based on the logic that the lower layers provide functionality to support the higher layers [30]. The order of the layers from a bottom-up perspective is: (1) Infrastructure Layer, (2) Application Layer, and (3) Business Layer. Further analysis on the layers and their aspects is provided in the sub-section regarding ArchiMate®.

2.4.2. Architecture Frameworks: Goals and Inputs

A variety of Architecture Frameworks (AFs) covering EM have been developed. Tang et al. [27] mention that AFs have improved the understanding of the EA concept by "providing systematic approaches to architecture development". It is also stated that AFs use viewpoints to create views that represent different perspectives of a system model. Furthermore, AFs assist structuring of architecture techniques by relating the viewpoints and the modeling techniques linked to them [31]. These viewpoints comprise the three layers that were introduced in the previous section: the business architecture, information architecture and technical (application and infrastructure) architecture.

Among the different AFs that have been developed over time, there are some underlying common goals. These goals as described by Tang et al. [27] are presented in Table 5.

• **Architecture Definition and Understanding** • **Architecture Process** • **Architecture Evolution Support**	• **Architecture Analysis** • **Architecture Models** • **Design Tradeoffs** • **Design Rationale**	• **Standardization** • **Architecture Knowledge Base** • **Architecture Verifiability**

Table 5 - Architecture Frameworks Goals

Since the goal of this research is to provide a methodology that will assist and improve impact analysis of EA models, the most relevant goals, which the outcome of this research can enhance, are:

- *Architecture Analysis*, which denotes the set of viewpoints for analyzing information in order to make architecture choices.
- *Design Tradeoffs*, which includes the selection of a design among different design choices by dealing with conflicting requirements.
- *Architecture Verifiability*, which incorporates the notion of providing sufficient information in the architecture design for review and verification.
- *Design Rationale*, which includes the documentation of the reasons behind design decisions for verification.

Additionally, AFs accept as inputs a variety of information from different sources in order to produce the EA representation. The input categories identified by Tang et al. [27] are presented in Table 6.

• **Business Drivers** • **Technology Inputs** • **Business Requirements**	• **Information System Environment** • **Current Architecture** • **Non Functional Requirements**

Table 6 - Architecture Frameworks Inputs

In this research, the focus is on quantifications derived from the business needs, so, while all input types are useful, special emphasis is put to input coming from the environment. Thus, the definitions for the following types of input, as stated by Tang et al. [27], are provided:

- *Business Requirements*, which include users' requirements, functional requirements, data requirements and other business system related requirements
- *Information System Environment*, which contains budget, schedule, technical constraints, resources and expertise, organization structure, other constraints, enterprise knowledge base.

2.4.3. ArchiMate®: Language and Metamodel

Enterprise modeling (EM) is another aspect of EA. Based on the notion of viewpoints which is essential for the concept of AFs, EM describes the EA from different viewpoints [29], since models are the actual means for transferring the EA into a comprehensive representation. EA models are responsible for specifying the system in detail [29], expressing the architecture as clearly as possible to the various stakeholders regarding their own understanding and communicating it to others [31]. Thus, models can be seen as languages that describe the EA and, ideally, all three different layers of EA (business, application, and infrastructure). According to Jonkers et al. [30], while modeling was supported separately within each of the three layers, there was no well-described concept which could sufficiently describe the relationships between the layers. Representing EA in layers is considered the essential way for capturing the internal structure of the elements of which the organization and business context consist, the elements' behavior as well as the information that is exchanged and used by the elements [30]. Because of the need for a coherent model relating all three layers, the ArchiMate® modeling language was developed. Such a language can support the modeling of both the structure within each domain in an easy-to-understand way for all stakeholders, and the relations between the domains [32].

ArchiMate® language should be applicable for a limited modeling purposes but non-specific enough, thus it consists of basic concepts which are considered the core elements of the language and which can enable further extensions of and additions on it [31]. ArchiMate® 1.0 focuses on the modeling of the 'extensional' aspects of the organization, which cover the external and internal properties of the organization in order to support its operational character [8]. The products of the enterprise, their value and the services that the products offer can be categorized in the external properties, while the internal properties include the processes, applications and the underlying technology that realize the services [8] [7]. The basis of the model lies on structuring the EA in two orthogonal dimensions: layers and aspects [7].

The *layers* dimension has already been introduced in previous sections, but a more complete description is offered in this section. Thus, based on the ArchiMate 2.1 Specification [17], the three core layers are the following:

- **Business Layer.** It includes the products and services that external users can access and which are offered by internal business processes and performed by business actors. As every layer has its own concepts in the language, the Business Layer covers concepts relevant to the business domain, such as business products, objects and services, business actors, interfaces, value of products and meanings of objects.

- **Application Layer.** This layer provides support to the business layer elements through application services which are actually realized by software applications. Some of the main concepts of this layer are derived from already existing standards for software applications descriptions, such as application service, interface, function and data object.
- **Technology/Infrastructure layer.** It offers infrastructure services (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication hardware and system software. In this layer, similarly with the application layer, the concepts have been adopted from existing standards. Some of the relevant concepts regarding technology are infrastructure service, function, interface, device, network, system software.

The *aspect* dimension is composed of the following modeling aspects [7]:

- **Structure aspect.** In this aspect, the actors which are involved in each layer are represented and the relationships among them are also modeled. Under the term 'actor', systems, components, people, departments, business units, etc. can be considered.
- **Behavior aspect.** This aspect represents the performed behavior by the actors and how the actors interact. Processes and services can be considered as behaviors.
  The behavioral concepts in each of the three layers (or domains) can be classified as either *internal* or *external* behavioral elements. [17]
    - **External behavior** represents the visible functionality of each layer which offers valuable information to the upper layer or the environment. The external behavioral elements are responsible for hiding the way the functionality is realized in the layer and therefore assuring the independence of the layers to some extent. The most common external behavioral elements are the 'service' and 'interface' elements of each layer.
    - **Internal behavior** elements are the ones responsible for realizing the functionality (services offered) and providing input to the external behavior concepts. Such elements are the business actors, application functions, infrastructure functions, etc.
- **Information aspect.** The representation of the knowledge regarding the problem domain which is used by the actors and which is exchanged in the interactions between the actors through their behaviors is the core contribution of this aspect.

The above mentioned layers and aspects are illustrated in Figure 4. In addition, in Figure 4 the modeling domains are shown. Each domain represents a set of concepts used in order to describe a perspective on the enterprise as viewed by one or more stakeholders [7].

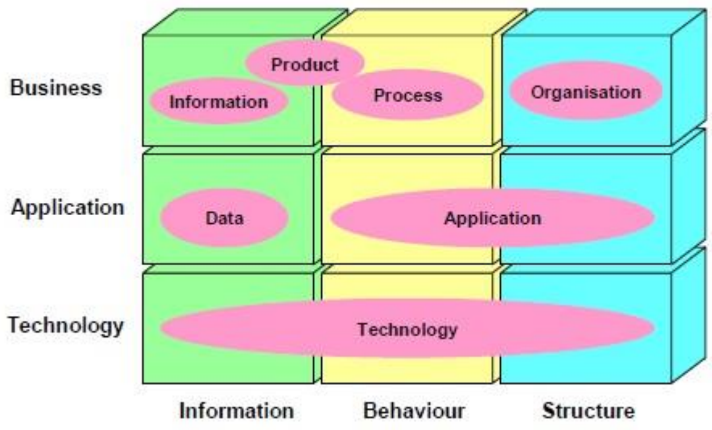


Figure 4 - ArchiMate® modeling framework, obtained from Quartel et al. [7]

As it is stated in the description of the layers, in every layer a group of concepts are used for modeling the functionality of the layer and a set of intra- and inter-relationships. The complete metamodel of the ArchiMate® language with all the concepts and the relationships between them is illustrated in Figure 5. Further information and more detailed explanations on these can be found in the ArchiMate Specification [17].

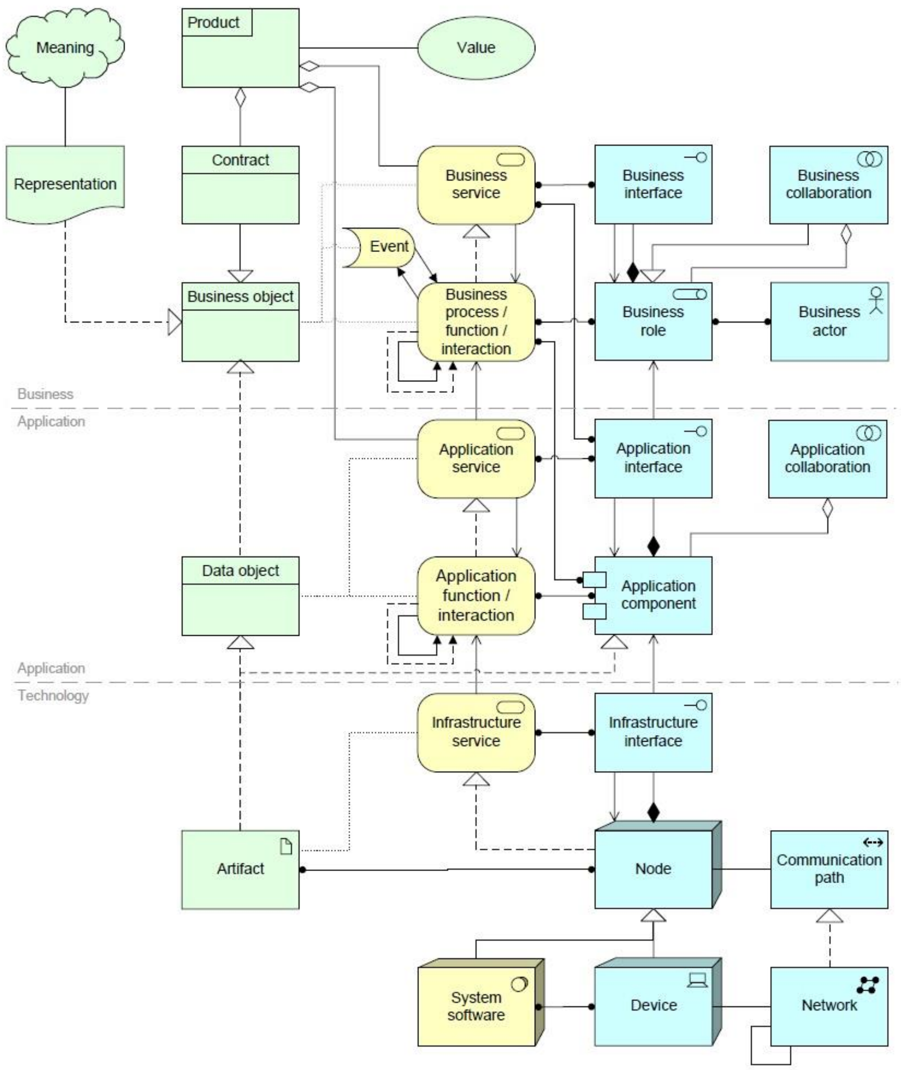


Figure 5 - ArchiMate® Metamodel, obtained from Buuren et al. [33]

2.4.4. ArchiMate® Motivation Extension and ARMOR language

Extensions of the ArchiMate® language have been introduced in order to capture in a better way the purpose of the language, which is to provide both extensional and intentional interpretations of the organization properties. The orientation of the extensions relevant for this thesis is towards providing explanations for modeling the *intentional* properties, i.e. *how* services are offered [7]. Regarding the layers, the added layer is the *Value* layer, which represents the value of the offerings (services, products) to the customers, while the additional aspect is the *Motivation* aspect.

The **Motivation aspect** is a core concept for this research as it covers the reasoning behind the existence of the architecture core elements, their contribution and in general the design of an EA model. It includes the actual motivations and intensions, represented by *goals*, *principles*, *requirements* and *constraints*, and the sources of the intentions, which include the *stakeholders*, *drivers* and *assessments* [17].

According to Chen et al. [29], EA is a way of communicating among stakeholders their expectations in terms of characteristics of the enterprise system without referring to detailed documentations of functions, data or resources. Thus, EA can be seen as a means for addressing the concerns and requirements of the stakeholders and the potential conflicts among them, and as an essential part of the requirements analysis process, since without EA, as Chen et al. mention [29], it would be "highly unlikely that all the concerns and requirements will be considered and met". Therefore the Motivation aspect provides the direct representation of the goals, concerns and requirements of stakeholders on the EA model. As it is illustrated in Figure 6, covers all four layers, including the Value layer, where the value of the architecture to the stakeholders and the necessary principles reside.

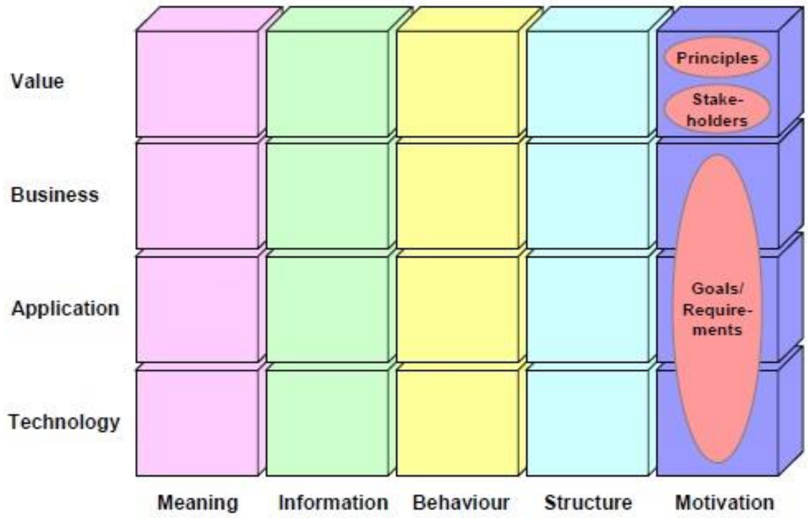


Figure 6 - EA Modeling Framework, obtained from Quartel et al. [7]

The basis of the Motivation extension is the ARMOR language which is introduced by Quartel et al. [7] and further refined by Engelsman and Wieringa [4]. ARMOR, initially, contained more concepts and relationships, such as a differentiation between soft and hard goals, refinement relationship and conflicts between goals [7], which were dropped for simplicity reasons and for improving the usefulness of the language from the customers' perspective. The simplified version of ARMOR is called Light ARMOR and it is described by Engelsman and Wieringa [4] as well. This language was developed by combining concepts of well-known GORE languages and modeling them in accordance to the EA needs. It has been also implemented in the BiZZdesign Architect tool. The concepts and the relationships of the Motivation aspect as well as their definitions are presented in Table 7.

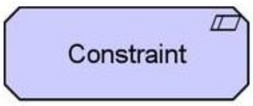
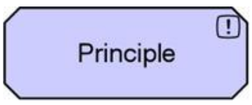
Motivational Concepts		Intentional Relationships	
Notation	**Definition**	**Notation**	**Definition**
	The role of an individual, team, or organization (or classes thereof) that represents their interests in, or concerns relative to, the outcome of the architecture.		***Association*** models that some intention is related to a source of that intention.
	Something that creates, motivates, and fuels the change in an organization.		***Aggregation*** models that some intentional element is divided into multiple intentional elements.
	The outcome of some analysis of some driver.		***Realization*** models that some end is realized by some means.
	An end state that a stakeholder intends to achieve. (see also Section 2.1.1)		***Influence*** models that some motivational element has a positive or negative influence on the realization of another motivational element.
	A statement of need that must be realized by a system. (see also Section 2.1.3)		
	A restriction on the way in which a system is realized.		
	A normative property of all systems in a given context, or the way in which they are realized.		

Table 7 - Motivational extension: Concepts and Relationships, based on ArchiMate 2.1 Specification [17]

Since the most significant contribution of the Motivation extension is its ability to justify the modeling of the core elements in the EA model, the linkage between the two levels of modeling should be emphasized. The connection is only provided through a 'requirement' or a 'constraint' concept which can be directly related to a core element of any of the three layers through a realization relationship [17], as it is depicted in Figure 7.

Except from the concepts introduced by the Motivation extension, new viewpoints are also available. Viewpoints define which modeling elements can be used as well as the allowed relationships between them, in order to present the view of different stakeholders depending on the need or concern that should be addressed and to facilitate discussion and understanding through various representations of the model. So, viewpoints can be used for isolating certain aspects of a model and for relating two or more aspects [17]. The most important viewpoints assigned to the Motivation extension, which can contribute to the determination of the connection between the goal level and the architecture level are the following as described in the ArchiMate 2.1 Specification [17]:

- **Goal Realization Viewpoint.** In this viewpoint the refinement of the high-level, business goals into more concrete goals which in turn are refined into requirements or constraints can be modeled. In this viewpoint the degree of decomposition of high-level goals and the relationships among different goals and requirements is important for understanding the interdependencies of alternative options or possible conflicts. Additionally, in case a requirement or constraint is related to more than one goal, the value of this requirement to different goals and the variability among these values can also be addressed.
- **Goal Contribution Viewpoint.** This viewpoint is a variation of the previous one. The difference is that it allows the modeling of the influence relationships as well. The degree of influence, either positive or negative, is expected to play an essential role in determining the value of a requirement and a goal.
- **Requirements Realization Viewpoint.** In this viewpoint, modeling the realization of the requirements from core elements of the architecture model is possible. As it is already mentioned, the requirements can be realized by elements from all three layers (Figure 7). This viewpoint is important because it enables the representation of the connection between the goal model and the architecture model which is valuable for the transition of the values assigned to elements in the architecture level to the motivation concepts. Furthermore, as Quartel et al. state [7], the assessment of strengths, weaknesses, opportunities or threats due to changing existing goals or defining new ones could be also facilitated through this viewpoint. Finally, a trade-off analysis can be enhanced by illustrating different realizations of the same goals in the architecture level.

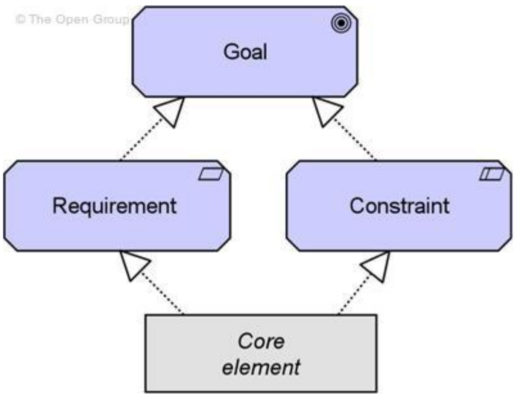


Figure 7 - Requirements Realization Viewpoint, obtained from ArchiMate 2.1 Specification [17]

Except from the connection with the EA model concepts, the motivational concepts can also be related with the 'Plateau' and 'Deliverable' concepts form the Implementation and Migration Extension [17]. These concepts denote, respectively, a relatively stable state of a Transition or a Target Architecture, and a specific outcome as part of a series of actions which aim to accomplish a specific goal. Through this linkage, modeling the intentions of the actions that comprise the intermediate stages of a future architecture and lead the process towards this transformation is possible (Figure 8).

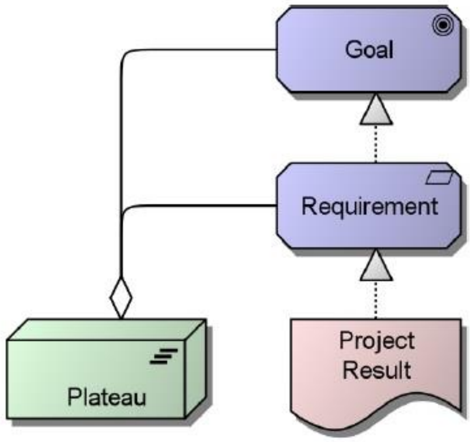


Figure 8 - Link between the Motivation and the Implementation & Migration Extensions

2.4.5. BiZZdesign Architect

BiZZdesign Architect is an EA software tool developed by BiZZdesign. It is a modeling tool that supports the design, visualization and analysis of EA on the three architecture domains [34]. According to IFEAD [35], Architect is a complete, integrated, effective and to-the-point solution which combines best practice models and methods. BiZZdesign Architect is based on the open standards of ArchiMate® and TOGAF, while it supports other AFs as well, and supports the analysis and demonstration of change impact through highlighting relationships and dependencies among concepts by following traceability links. The comparison between different instances of the model as well as the isolation and generation of parts of the model by selecting the elements of interest is also possible. This functionality is of interest in this Thesis, since the generation of the sub-graphs that span the motivational modeling level and the architecture is essential for the quantifications.

2.5. Summary

This chapter provided the explanation and description of the basic terminology and the essential concepts that support the research in this thesis. As it was determined in Chapter 1, the goal of this research is to assess the value of EA in the motivational level and the architecture level. Thus, the establishment of the terms and the definitions, which concern both levels, as well as concepts that will contribute to the design of the quantification mechanism(s), such as traceability, was accomplished. In the next two chapters, the existing techniques that can assist the quantifications and analysis in each modeling level are be presented and described.

CHAPTER 3

3. EA Model Analysis Techniques

This chapter comprises an exploration regarding the techniques that are suitable or related to the EA model analysis, and addresses the second phase of the Design Cycle, i.e. the Treatment design. The techniques included in this chapter address cost and financial analysis, benefit identification and performance measurements.

3.1. Quantitative Analysis

Quantitative analysis is a procedure in which at least one model is created in order to analyze possible architectural solutions and alternatives, and evaluate their quality [36]. The analysis performed on these models produces quantifications as results, which contribute to the architectural knowledge. The term 'architectural knowledge' includes the knowledge that is produced as an output from the EA analysis process and which is used as input for future analyses. Having and sharing the pool of knowledge resulting from quantitative analysis, which can be continuously enhanced through iterations of EA analysis, contributes in, according to Jansen et al. [36]:

- the *integration* of analysis performed in sub-models of the architecture
- the *evaluation* of the system as a *whole*
- performing a more complete check for *correctness*, *completeness* and *consistency* of models and the system that are analyzed
- the *validation* of an architecture design by supporting the rationale behind a decision for a specific design. Validation is also accomplished with the help of concepts such as traceability, discovery and trial of alternative solutions, trade-off analysis.

According to Jansen et al. [36], the quantitative analysis process includes a set of concepts, presented in Figure 9. A short description of the notation and the relationships between the concepts is provided in Table 8.

Concept	Description
Alternative	*Alternatives* are different design options.
Design Concept	*Design Concept* is a classification of Alternatives. A Design Concept is the basic type of design, it provides a solution direction and defines the scope in the design space.
Scenario	A Design Concept is specialized in one or more *Scenarios* which are the target of the analysis.
Analysis Model	It is an aggregation of *System Parameters*, *Analysis Functions*, *Quality Attributes* and *Numbers*.
Analysis Function	Describes a System Parameter's behavior and relationships.
System Parameter	- Describes the state of part of the Analysis Model. - Is the input and output of the Analysis Function. - Defines the quantified unit.
Quality Attribute	Defined by the quality model(s) used in the analysis.
Value	*Value* represents the assigned value to the input System Parameter of the Analysis Model.

Analysis Output **Number**	*Analysis Output* represents the output of the Analysis Model calculated by the Analysis Functions. Both of them are specializations of *Number*, which represents the units defined by the System Parameter.
Confidence	*Confidence* is a property of an Analysis Function. It is not designed in the model in Figure 9, but it is important since it indicates the reliability of the Analysis Functions. The reliability depends on the degree of the functions relying on intuition, trends, facts, or guesses.

Table 8 - Quantitative Analysis Process Concepts & Description

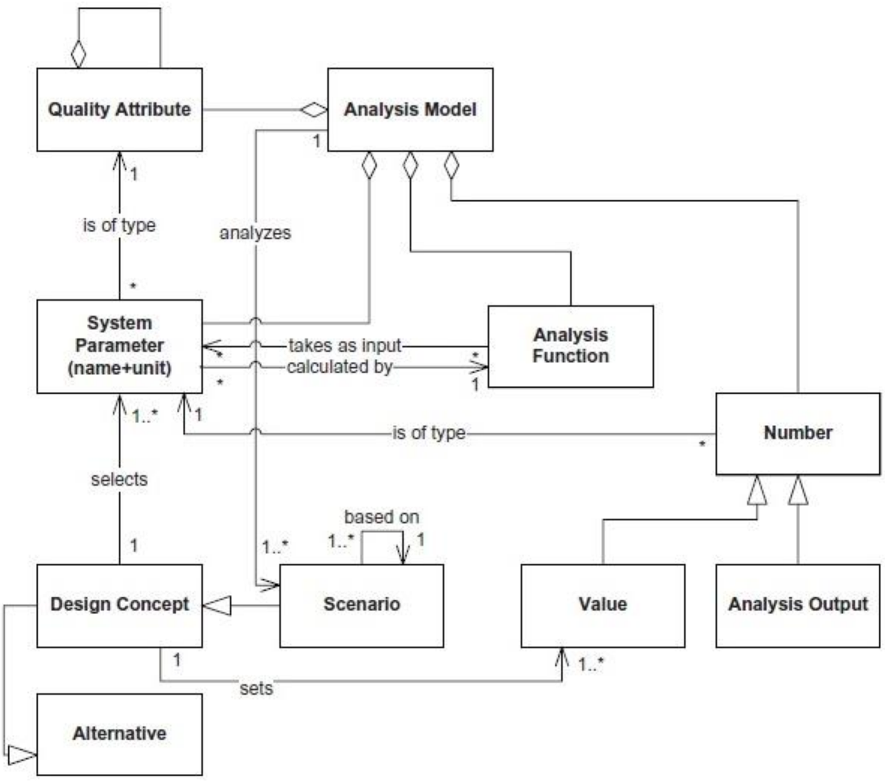


Figure 9 - Quantitative Analysis Process Concepts, obtained from Jansen et al. [36]

The concepts of this model will be used as guidelines for developing and designing the quantification mechanism(s) in this thesis. Based on the notation of the quantitative analysis process concepts, such a quantification mechanism will comprise an aggregation of Analysis Models and Functions, i.e. the quantitative analysis techniques discovered in literature and practice. These techniques will contribute to the design of the mechanism and will provide the actual calculations (outputs) of the analysis.

In this Chapter as well as in Chapter 4, the techniques, which will support the calculations and measurements of different parameters of the EA and goal models, are identified and described. The parameters and the quality attributes, that will be assessed, vary due to the two levels of modeling, i.e. architecture model and motivational or goal model. This Chapter focuses on the architecture analysis, where financial measures and performance measures address the needs of the three architecture layers, i.e. business, application and infrastructure.

Before moving to the description of the techniques, a concern presented by Abran et al. [37] should be addressed. According to Abran et al., quantifications lack the rigor of measurement which is provided by the rigor of metrology. In the case of this thesis, the quantifications will be based on actual measurements and not on qualitative estimations. In that way the rigor of the mechanism will be guaranteed.

3.2. Transformations, Analysis and Aspects of EA

The quantitative analysis of EA models falls into the model analysis activity. Analysis is necessary to ensure that a design of the architecture is aligned to the requirements, either functional or non-functional, and/or to enable the performance of optimization checks [38]. To perform the analysis, though, it is often a prerequisite to obtain a specific analysis model which is representative of the design model. An analysis model is usually expressed in a separate analysis language which indicates the correspondence between the design model and the analysis model [38]. The corresponding analysis model is the result of the transformation on the design model. There is one type of model transformations that is of importance for this Thesis:

- *Horizontal model-to-model transformations*. This type of transformations handles the transition from a model in the design space to a model in the analysis space. This process usually includes the initial transformation from the design space to the analysis space and a reverse transformation from the analysis to the design space when the quantifications have been accomplished [38].

Based on these observations and by taking into consideration that ArchiMate® is a modeling language that can be placed in the design space, both the architecture model and the motivational model belong to the design space. To be able to perform the quantitative analysis, the need of a suitable transformation of the models into more appropriate analysis models for applying the analysis techniques has been identified. In Figure 10, the proposed model transformations are illustrated. The arrows on the side indicate the relationship between the two models. In the design space, the link between the goal model and the architecture is established through ArchiMate®, while in the analysis space a predefined relationship does not exist (which is the reason for the dashed red line). During the development of the quantification mechanism(s) in this thesis, an attempt for determining and specifying this relationship is also included.

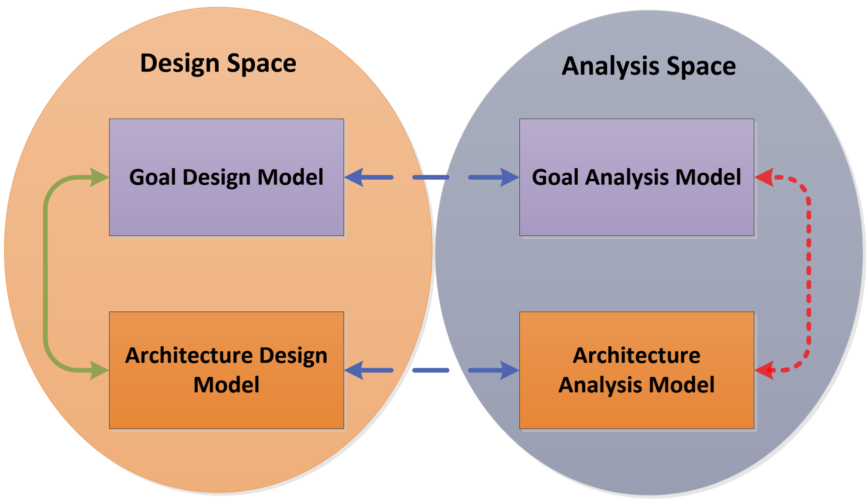


Figure 10 - Design & Analysis Space: Transformations

Another important viewpoint of the EA analysis is related to the cross-cutting aspects of EA as presented by Matthes et al. [39] and Buckl et al. [40]. Except from the areas that are directly related to EA, i.e. the business, application and infrastructure architectures, there are a number of cross-cutting aspects that surround this topic (Figure 11). The four identified cross-cutting aspects, according to Buckl et al. [40], are:

- *Strategies & Objectives*: This aspect concerns the business and IT strategies and their operationalization by goals and objectives [39].
- *Requirements & Projects*: In this aspect, the requirements and further decomposition of the necessary actions for the operationalization of the strategies and the goals are included [39].
- *Blueprints & Patterns*: This aspect concerns the reference models or blueprints which are used as patterns for increasing the standardization in a company [39].
- *KPIs & Metrics*: This cross function provides the way for quantifying EA aspects by assigning metrics and measures to information objects. The assignment is possible from either the other layers of the architecture or the cross aspects and as a result it offers a set of indicators for various properties. The cross-cutting concept of measure assignment can extend the quantifications of EA elements [39].

In the context of this thesis, the relevant cross-cutting aspects are indicated with the orange color in Figure 11. Firstly, in the 'Strategies & Objectives' and 'Requirements & Projects' aspects, the identification and refinement of goals and projects that guide the current EA design and the transition to a to-be architecture are included. Secondly, in the 'KPIs & Metrics' aspect, the desired quantifications that support the analysis and the decision-making regarding the design are incorporated. Hence, the cross-cutting functions of EA provide the notion for linking the quantifications to both EA and the strategic and goal level. Additionally, the structure proposed in Figure 11 demonstrates the direction of the relation between the metrics and the goals. The measurements are directly assigned to EA elements and then transferred to the goal level, since the EA layers serve as the facilitators of the transfer, even if the rationale for selecting the measures and the indicators is provided by the goals. Further clarification of this transfer of measures and indicators is also part of this thesis.

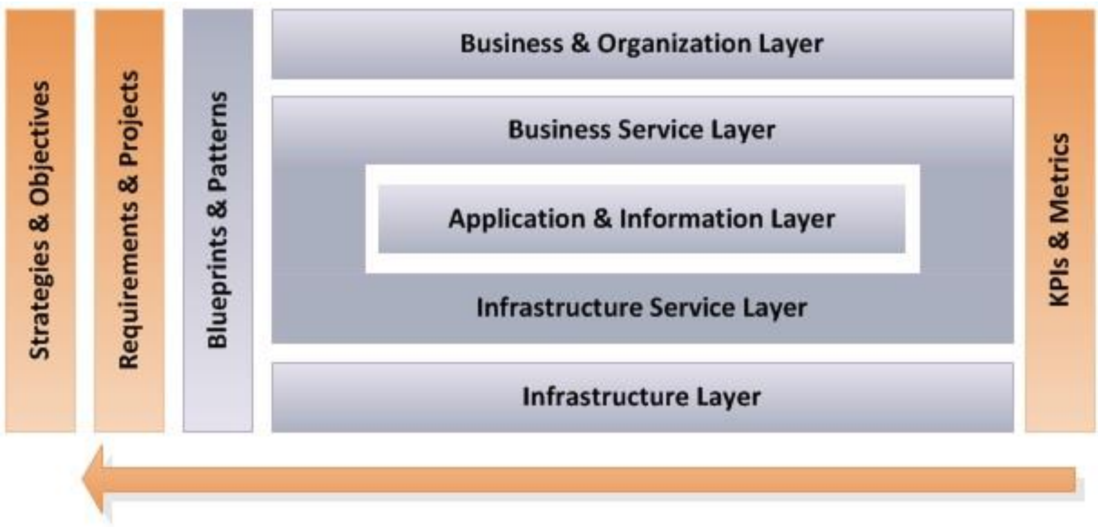


Figure 11 - Cross-cutting aspects of EA, obtained from Buckl et al. [40]

3.3. Costs and Benefits in Financial Analysis

3.3.1. What is 'Cost'?

Cost is a basic term which needs to be clarified and defined in order to establish its use in this thesis. In literature many different types of costs have been identified. Additionally, a variety of cost factors has been assigned to or associated with each cost type. An overview of the cost types and the associated cost factors is provided in this section.

Schwetje and Vaseghi [41] make a distinction between the terms 'investment' and 'costs':

- *Investment* is the required capital. There are different types of investments, such as: *material* investments which include machinery, *financial* investments such as shares, *immaterial* investments such as computer software, and *replacement* investments.
- *Costs* are all the remaining expenses: salaries, wages, office supplies, information processing, telecommunication, etc.

According to Mutschler [42], a general definition of *costs* covers the "total expenses for goods or services including money, time and labor". Examples of costs associated with EA are given by Schekkerman [43]. Schekkerman views the costs of EA as an accumulation of the expenses related to activities, services, people, labor, training, tools, verification and validation, and compliance and maturity assessment. Thus, except from money, time and labor, as another general category, resources can be added.

It can be observed that the term 'expenses' is also used in relation to costs. In some cases there is an interchange of the terms, as one is used to explain the other, but most commonly *expenses* can be seen as a broader term which represents an outflow of money to a third party to pay for an item or service or for a category of costs, according to Wikipedia [44].

While costs cover money, time, labor and resources, there are many different classifications of the costs depending on the aspect that needs to be emphasized each time, e.g. variability, business activity-related. Moreover, vast research can be found on costs and cost factors related to IT projects and investments. Thus, in the cost types described in Table 9, special attention has been paid for specializing them for the IT sector. The cost factors that are associated with the cost types are described in more detail in Section 3.4.3. dedicated to Total Cost of Ownership (TCO).

Cost Type	Description
Acquisition costs	These costs cover the initial purchase costs and the finance costs: • **Purchase** costs for purchasing an asset, such as a system or an application, after adjustments for incentives, discounts, or closing costs, but before any sales tax [42]. They can also include the assessment of the items to be purchased [45]. • **Finance** costs include the expected effect (in terms of costs) of alternative sources of funds for acquiring the specific asset(s) [45].
Implementation costs	These costs include all the costs associated with the implementation of the project. Since, IT and EA projects refer mostly to equipment, the implementation can be decomposed in: • **Development**, e.g., of software application. In the case of EA, development costs can also include the costs for the implementation

	of the architecture functionality, the cost of fitting the architecture within an existing environment and the cost of connecting the business architecture to the existing business processes [46]. • **Installation** of applications or systems [45] • **Commissioning** of external consultants [45] • Initial **training** of the employees [45] Implementation costs can also be considered as the initial part/phase of the operating costs.
Initial capital costs	The two previous categories can also be found in literature as Initial capital costs. This is a more general term which covers "all the costs of buying the physical asset and bringing it into operation" [45]. The goal of the capital costs is "to create the capability" [47].
Operating costs	These costs are the costs necessary to run the capability [47] and they include periodic or recurring costs [48] [49]. Operating costs include a wide area of costs: • **Hardware costs** • **Software costs** • **Personnel costs** (IT staff and users [48]): these costs can be further categorized depending on the functional area of the employees [41]. They cover the labor costs as well as social costs, training costs. Considering the users of the system as a subcategory serves in determining more detailed costs, such as user troubleshooting and help [48]. A type of personnel costs are also the **Administrative costs** which are the costs for managers, accountants and in general employees who are not directly involved in the operations of the project, as well as their associated office costs [12]. • Important examples of **recurring costs** are: license fees, telecommunications, ongoing maintenance costs [49]. Vienneau and Nicholls [48] limit the recurring period to 3-5 years. Without this limit to be mandatory, it is a logical assumption since technology progresses very fast. Additionally, Woodward [45] mentions that the operating costs cover both direct and indirect costs, the estimation of which depends on predictions and experience with similar projects or assets.
Maintenance costs	As it is already mentioned, maintenance costs are a type of recurring costs. Maintenance is essential for sustaining the capability [47]. There are two different types of maintenance [45]: • *Corrective or unplanned maintenance*, which takes place when a fault occurs and may hinder high system downtime costs, and • *Preventive or planned maintenance*, which suggests scheduled maintenance of the systems leading to lower downtime costs and higher labor costs. Thus, maintenance costs include both labor costs as well as costs related to the equipment [45].
Ongoing Change & Growth costs	These costs take into account future costs associated with additions and changes of the technology environment [48]. Change and growth can affect the equipment as well as the personnel and the sites where the equipment resides.
Life Cycle Costs (LCC)	Life cycle costing refers to the costs that occur during the entire lifecycle of an investment [42]. The definition selected by Woodward [45] as the most useful and representative one is the following:

	"The life cycle cost of an item is the sum of all funds expended in support of the item from its conception and fabrication through its operation to the end of its useful life".
	Since LCC cover the entire period that a system operates, an appropriate selection of the cost structure is mandatory depending on the project or the system that is analyzed or estimated. The selection according to Woodward [45] has to suit the objectives of the project and the company that runs the project.
	A few important points regarding LLC are the following:
	• Determination of the *utilization factors* [45] of the system is an element of the LCC procedure. It can indicate the way a system will be functioning and consequently its life expectancy and the associated requirements for the selection of the system.
	• *Life of the asset* indicates the forecasted life which can be determined, for example, in terms of:
	    ○ *Technological life* [45]: the period until a technologically superior alternative is developed which would necessitate the replacement of the asset.
	    ○ *Economic life* [41] [45]: the period until economic obsolescence makes the replacement of the asset inevitable. The expected economic lifetime is also related to the monthly/annual depreciation of the asset.
	All previous cost types are included in the LCC (acquisition to change & growth costs). A final type of costs which is important for the LCC and related to the life of the asset is the **Disposal cost**. This cost occurs at the end of the life of the asset and it includes the cost of demolition, scrapping or selling the asset [45].
	Schekkerman [43] refers to the *EA Lifecycle Cost* as:
	$\boldsymbol{EA\ Lifecycle\ cost} = EA\ initiation/definition\ cost + EA\ development\ cost + EA\ implementation\ cost + EA\ maintenance\ cost$
	In this case, the initiation/definition costs are related to the analysis and evaluation of different designs of EA and the negotiations with the stakeholders until the EA project is approved.

Direct & Indirect costs	This classification aims at distinguishing between the costs that can be directly assigned to an activity, service or product and those which cannot. The indirect costs cannot be represented by explicit cost factors, such as depreciation or maintenance [42]. In EA, the direct and indirect costs refer to activities that are necessary for supporting the program of EA [43]. This logic is related to the activity-based costing method, which is described in Section 3.4.4.
Fixed & Variable costs	*Fixed* costs are the costs which remain approximately in the same level because they are independent of the production of a process, the output or the utilization of a service. Examples of fixed costs constitute rents and administration costs. [42] *Variable* costs, on the other hand, depend on the level of output. These costs correlate with the change in the level of input. Examples of variable costs constitute labor and commission costs. [42]

Table 9 - Cost Types

3.3.2. What is 'Benefit'?

Making a decision for realizing a change in an organization depends not only on costs. Costs provide a partial view of what a change carries with it and they particularly show the burden that it will be borne by the company for implementing it. In order to make a good, robust and secure decision, the assessment and analysis of the benefits that the change project will offer to the company is essential. Except from defining the benefits, a process should be in place for managing and organizing the potential benefits, and making sure that they will be realized [13].

Benefits of a project or an investment often resemble the goals or objectives of the investment. According to the methodology proposed by Ward et al. [49] and refined by Eckartz [50], objectives derive from business drivers and they represent what the proposed investment or project will provide to the company. Benefits, on the other hand, represent the advantages received as a result when the objectives will be met [49]. While goals may concern a specific group of stakeholders, benefits may be of interest of an even wider or different group of people. Thus, the expected benefits must be clearly identified and specified, which will also support the validation of the goals realization.

Benefits Identification

For the identification of benefits, Eckartz [50] highlights the benefits identification matrix as suitable for the initial phase of documenting the expected benefits. The classification of benefits in this matrix is across three dimensions (Figure 12).

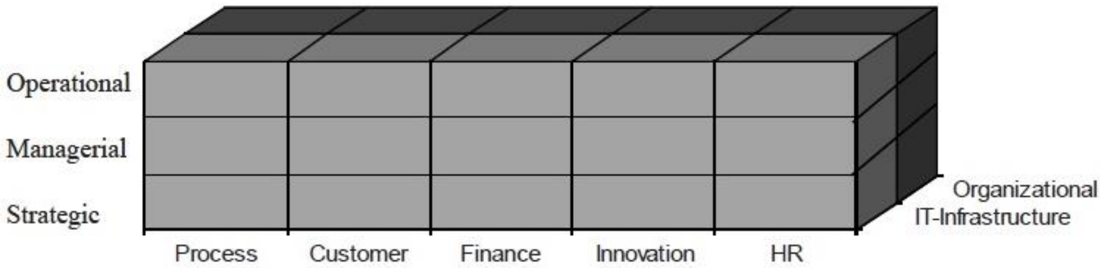


Figure 12 - Benefits Identification Matrix, obtained from Eckartz [50]

As it is argued by Eckartz [50], the financial benefits are of great importance for the senior management of the company, while 'soft' or intangible benefits are of great significance for the success of the project or investment. Eckartz also states that the activities related to innovation/HR and Organizational/IT-infrastructure are, on the one hand, crucial for implementing a project successfully, while, on the other hand, they include hidden costs due to the need of change management and training. In this thesis, since the goal is to quantify the benefits that will arise from a change in the EA, the focus will be on financial benefits and benefits that can be accurately measured.

An extensive research was conducted by Eckartz [50] for providing examples of benefits and benefit-indicators fitting in each of the blocks presented in Figure 12. APPENDIX I includes the detailed list of benefits and indicators identified by Eckartz [50]. Based on the identified benefits in each category, an observation can be made regarding their quantifiability. Benefits found in the operational and managerial level as well as in the IT-Infrastructure level are more tangible than the rest. Since these benefits are further grouped across the horizontal dimension categories, in most cases the benefits found in Process, Customer and Finance are more quantifiable. Table 10 presents the categories where tangible or quantifiable benefits are more likely to be found.

	Process	Customer	Finance	Innovation	HR
Operational	x	x	x	x	
Managerial	x	x	x		
Strategic					
IT-Infrastructure	x		x		
Organizational					

Table 10 - Where can quantifiable benefits be found?

Benefits Specification

Since benefits are seen as the realized advantages, the description provided by Schekkerman [43] regarding benefits functions as a set of examples: "monetization of increased operational efficiency, reduced operational costs and personnel numbers, increased customer satisfaction and consolidated information systems". But not all of these benefits can be directly or easily measured. Benefits such as the 'increased customer satisfaction' are more difficult to quantify and the availability of data is an important factor for determining the monetary value of it. For supporting the attempt of measuring benefits and transforming them from intangible to as tangible as possible, special emphasis should be placed on the methodology proposed by Ward et al. [49] and enhanced by Eckartz [50]. The core characteristic of this methodology is the classification of benefits in four categories based on the ability to measure them or, in other words, their 'degree of explicitness'. The four categories are presented and described in Table 11, which is based on their definition by Ward et al. [49].

Degree of explicitness		
High	**Financial benefits**	1. *Measure* for financial value: application of a cost/price or other valid financial formula to a quantifiable benefit. 2. Sufficient data for verifying that the benefit can be achieved. 3. Combination of the calculations provides an overall financial value.
	Quantifiable benefits	1. *Measure*: calculation of an existing or easy-to-implement measure based on reliable estimation of the expected size of the benefit. 2. Forecasting techniques for providing a realistic economic value.
	Measurable benefits	1. *Measure*: calculation of an existing or easy-to-implement measure. Existing measures as the ones already used for the organization performance or the existing key performance indicators (KPIs). 2. Not easy or possible to make certain estimation about the performance improvements after the change is achieved. 3. Qualitative-oriented measures
Low	**Observable benefits**	1. *Measure*: assessment of the degree of benefit realization based on experience, opinion or judgment and a set of predetermined criteria 2. Lack of historical data. In case historical data are available, the benefit would be considered 'measurable'. 3. Subjective, intangible or purely qualitative measures

Table 11 - Benefits classification based on Degree of Explicitness

Three important additions to the classification presented in Table 11 are the characteristics of the *time span* of a benefit, i.e. when the benefit will occur, the *probability* of achieving the benefit, i.e. the probability for achieving its expected results, and the *frequency* of achieving the effects of the benefit in a specific time span [50]. These additional characteristics provide a more complete description of a benefit, enhance the comparison of benefits and support a more accurate and realistic estimation of their economic value. In the discussed models ([49] [50]) there are characteristics assigned to benefits (Figure 13), which are considered as out of scope for this thesis. The current focus is on the 'Measurement of effect', and especially on the top three categories (financial, quantifiable, measurable), and on the 'time span, probability and frequency' of benefits. The 'Classification of change' could also be of use.

Benefit	**Benefit owner:**	**Classification of change**	**Required business changes**	**Measurement of effect**	**Time span:**
		Do new things (grow the business):	Process level:	Financial:	**Probability:**
	Subject matter expert:	Do things better/ cheaper/faster:	People level:	Quantifiable:	**Frequency:**
		Stop doing things:	Organizational level:	Measurable:	
			Technology level:	Observable:	

Figure 13 - Benefits Specification Matrix, obtained from Eckartz [50]

3.4. Economic-driven IT evaluation approaches

One of the basic aspects when evaluating and analyzing a potential investment or project is its financial value. In literature, a wide variety of techniques can be found and has been developed across the years for evaluating the economic viability of investments. In this section, the more common techniques for the evaluation of IT investments, and consequently for EA projects, are described.

Mutschler [42] has provided an assessment framework and a classification of the approaches regarding cost estimation and evaluation of IT investments. The proposed grouping of the approaches has been used as a guideline for the identification of cost evaluation and estimation techniques in Sections 3.4, 3.5 and 3.6.

3.4.1. Static Measures

Static measures are the ones that do not take into account the time value of money. The value of money decreases over time [12] which is expressed by the use of the 'discount rate'. Discount rate is a factor that quantifies the 'time value of money' [51] and determines how more valuable is the money today comparing to the future. In the static measures, though, this factor is not included, which functions as an advantage for them due to the simpler calculations [51]. Thus, the cash flows calculated are not discounted to the present value of money.

Return On Investment (ROI)

Explanation. ROI is the measure of the profitability of the investment [12] and it indicates the effectiveness of generating profit by spending money. It is a ratio of the profit to the total cost [42] showing how many times the profit of the IT investment (benefits minus costs) is higher than the capital spent for the investment.

Formula.

$$ROI = \frac{benefits - costs}{costs} * 100\%$$

Advantages. ROI is a relative measure [12], i.e. it does not depend on the size of the investment, and therefore it allows the comparison of projects with different sizes. Moreover, due to the simple calculation, it can be easily turned into a discounted measure by using the present value of benefits and the present value of costs.

Disadvantages. There is no defined limit for determining an acceptable value of ROI [12]. The determination of the limit depends on the stakeholders.

Payback Period (PP)

Explanation. The payback period is the time (in years usually) it takes for a project to cover the initial investment [51]. Thus, PP is the point in time where the cumulative cash flows generated each year are equal to the initial investment. It is common sense that the shorter the period in which the initial investment is returned, the more attractive the project is [12].

Formula.

$$Cumulative\ cash\ flow = Investment$$

Usually, the cash flows are calculated in an annual fashion and the cumulative cash flow turns form negative to positive from one year to another. To make a more precise calculation of the PP, a linear interpolation can be used [12]:

$$y = ax + b$$

$$0 = \frac{CF^{+} - CF^{-}}{year^{+} - year^{-}} * (PP - year^{-}) + CF^{-}$$

where:

CF^{+} is the cash flow for the year where CF >0 $year^{+}$ is the year when CF>0

CF^{-} is the cash flow for the year where CF <0 $year^{-}$ is the year when CF<0

Advantages. The payback period is easy to calculate, communicate, and understand [51]. Another advantage of PP is that it can be easily turned to a dynamic (discounted) measure. The only change is that the cash flows used for the calculation have to be first discounted to the present value of money [12]. In that way the result is called 'Discounted Payback Period'.

Disadvantages. It does not provide a specific value as a decision point for the decision-makers [12]. Generally, a short payback period is perceived as a way to avoid risks. Using this measure as the only one for the evaluation of an investment, though, will lead to arbitrary decisions [51]. Another drawback is the fact that IT projects with high benefits in the future may be rejected due to the long expected time [42].

Accounting Rate of Return (ARR)

Explanation. The accounting rate of return is another indicator of the profitability of the investment. Additionally, it considers the accounting net income rather than the cash flows by considering the depreciation in the computation of the cash flow [51].

Formula. There are a variety of different ways to calculate the ARR measure based on the way the benefits and costs are perceived, e.g. considering tax or not, considering the whole investment or an average value of it, etc. [51]. An indicative formula would be the following [51]:

$$ARR = \frac{average\ income}{average\ investment}$$

Advantages. The general form of the calculation enables the expression of the measure in a discounted fashion as well. Moreover, it takes into account the entire lifetime of a project [42], which makes this measure a better indicator when LCC is considered.

Disadvantages. Not using the cash flows for the calculation of the measure can be considered a drawback. Additionally, there is no established standard for evaluating how good the rate can be leading to arbitrary decisions [51].

Break Even Analysis

Explanation. The break-even analysis determines the point where the total revenues from the IT investment equal the total costs [42] based on fixed and variable costs. This method belongs in the evaluative quantification techniques [52], which aim to evaluate and compare alternative options. The calculation of the break-even point can be either in units or in total costs (money) and it requires estimations of the revenues, fixed and variable costs [52]. It is possible to use this method for calculating the break-even point of a single project or service and for multiple. Additionally, it can be used for evaluating alternatives by assessing the relative merits of two options and identifying the point where the two alternatives have equal total costs, i.e. the decision point [52].

Formula. The formula selection depends on the desired outcome and comparison:

1. Break-even point for a single product/service

   Formula for break-even point in *units* [52]: $BEP_x = \frac{F}{P-V}$

   Formula for break-even point in *costs* [52]: $BEP_€ = \frac{F}{1-(V/P)}$

   where:

$\boldsymbol{BEP_X}$: break-even point in units	$\boldsymbol{BEP_€}$: break-even point in euros
$\boldsymbol{P}$: price per unit after discounts	$\boldsymbol{F}$: fixed costs
$\mathbf{x}$: number of units produced	$\boldsymbol{V}$: variable costs
$\boldsymbol{P*x}$: total revenues	$\boldsymbol{F + Vx}$: total costs

2. Break-even point for multiple products/services: when the company has a variety of offerings with different price and variable cost.

$$BEP_€ = \frac{F}{\sum\left[\left(1 - \frac{V_i}{P_i}\right) * W_i\right]}$$

   where:

   $\boldsymbol{F}$: fixed costs
   $\boldsymbol{V}$: variable costs
   $\boldsymbol{P}$: price per unit after discounts
   $\boldsymbol{W}$: percent of each product/service in relation to total sales
   $\boldsymbol{i}$: each product/service

3. Evaluation of alternative processes: the result of the formula is the point where one alternative outweighs the other.

$$x = \frac{F_a - F_b}{V_b - V_a}$$

   where:

   $\boldsymbol{a}$ and $\boldsymbol{b}$ are the two alternatives and $\boldsymbol{F}$ and $\boldsymbol{V}$ the fixed and variable costs respectively.

Advantages. It enables the comparison of alternative solutions. Its expressiveness in units and money enables its use in a broader amount of cases and fields.

Disadvantages. It is dependent on the fixed costs. The higher the fixed the costs, the higher the break-even point [12]. On the other hand, depending on the fixed costs can serve as an indication for reducing the fixed costs in order to achieve less risk.

3.4.2. Dynamic Measures

Dynamic are the measures which take into account the time value of money. Thus, in the calculations, the cash flows are discounted. The discounting rate adjusts the returns for two perspectives. Firstly, the time value which is the inability of having the resources available for a period of time and, secondly, the degree of risk of not receiving the cost savings or expected benefits as planned [51]. Determining the risk is possible by establishing the variability of the returns of a project, an activity or a process. The variability can be established by using a Standard Deviation and a coefficient of variation [51].

Net Present Value

Explanation. The net present value is the technique for discounting all the costs (consumed cash flows) and benefits (generated cash flows) of a project to the present value of money [42] [12]. It is calculated as the discounted benefits less the discounted costs.

Formula. Two formulas are provided, a simplified [51] and a more analytic and representative one [12]:

$$NPV = PV - I$$

$$NPV = \sum_{t=0}^{n} \frac{CF_t}{(1+k)^t}$$

where:

$\boldsymbol{PV}$: present value of the benefits

$\boldsymbol{I}$: the investment

Advantages. It is an absolute measure, which means that it depends on the magnitude of the contribution of the project to the profit of the company [12]. It can assist in defining the set of projects that maximize the total value of the company [12]. NPV is an easily interpretable measure, as it represents the amount of value added to the company by the project and is accepted when NPV>0 [12]. It is also additive; hence the total NPV of a number of projects is just the sum of the individual NPVs [12]. Moreover, it supports the comparison among different projects, as NPV is a characteristic of the project itself [12] and it can be used for determining the most effective allocation of the budget [51]. It is also ideal for projects with complex return sequences, with different sizes and uncertain timings [51].

Disadvantages. The main concern arises from the decision on the appropriate discount rate and the level of uncertainty or inability of predictions for the future cash inflows and outflows. The highest risk and uncertainty is usually hidden in the predicted benefits (inflows). The sensitivity of the NPV to the discount rate can be determined by changing the discount rate and viewing the NPV as a function of the discount rate for given cash flows [12].

Internal Rate of Return (IRR)

Explanation. Internal rate of return is the discount rate when NPV equals to zero [12] [51]. It is the annual rate at which an investment is estimated to pay off [42]. IRR is considered a discounted or dynamic measure because it deals with the return of a project and its timing by considering expected life cycle cash flows [42] [51].

Formula. IRR is calculated by solving the equation:

$$NPV = 0$$

Thus the NPV formula presented above takes the following form:

$$0 = \sum_{t=0}^{n} \frac{CF_t}{(1 + IRR)^t}$$

Advantages. It is a well accepted measure from the executives, probably due to its rate of return form which is quite appealing especially because it can be directly compared to market rates of return [51]. IRR is accepted when it is higher than the market rate of return [51], thus there is a standard comparison limit.

Disadvantages. A variety of disadvantages are identified for this measure. Firstly, the equation may yield different results (more than one values for IRR) when the cash flows change from negative to positive more than once [51] [12]. Secondly, it is a relative measure and as it was explained, previously, it does not depend on the amount of the investment [12]. This can be a drawback when comparing projects with different sizes of returned value and initial investment because of neglecting these factors (scale problem) [51]. Finally, NPV and IRR may result to contradicting recommendations for projects with different cash flow profiles, as IRR may rank projects in an inconsistent way comparing to NPV [12]. Fisher's intersection is a way of overcoming the inconsistency or even identifying it from the beginning, as it provides interpretation of the conflict [12].

Profitability Index (PrI) or Benefit-Cost ratio

Explanation. The profitability index takes the present value of the cash inflows and scales it to the initial investment [51]. It is accepted when PrI > 1 [51]. The profitability index is the same as the Benefit-Cost ratio, which is the ratio of the present value of all the benefits to the present value of all the costs [12].

Formula. The calculation of the PrI is a ratio of the sum of the cash flows generated each year to the sum of the investments made each year [12]:

$$PrI = \frac{PV\ of\ cash\ generated}{PV\ of\ investment}$$

Advantages. It can be combined with the NPV as a secondary measure for ranking or prioritizing the alternative options when, for example, a budget has been predefined [51]. Thus, it can help in optimizing the allocation of the investment.

Disadvantages. There is a danger that the PrI may exclude projects with high NPV during the budget allocation and, thus, leading to inconsistencies in the ranking suggested by the NPV [12] [51]. Additionally, since it is also a relative measure, it has the scale problem, similar to IRR [51].

3.4.3. Total Cost of Ownership (TCO)

Explanation. Total cost of Ownership is a cost-oriented approach which is widely used both in research and in practice [53]. It is a technique that assists in assessing the effectiveness of the IT investments [54]. It constitutes a holistic view of costs, as its basic idea is that the organization must identify and consider all the costs of an IT product, from its purchase till its disposal [54] [55]. In that sense it resembles the Life Cycle Costing (LCC), since it captures the total changing costs of products throughout their lifetime [55]. One of the main notions in the research regarding costing is the need of managing and reducing the indirect costs, thus TCO is accountable for dealing with both direct and indirect costs [56].

The operationalization of TCO includes classifying costs in cost categories and then defining cost factors for each category. The most common cost categories related to the IT field are: (1) hardware costs, (2) systems and applications, (3) administration and administrative support, (4) operations costs, (5) control costs, and (6) the hidden costs [54] [55].

The control costs consist of centralization costs and/or standardization costs, which are complementary methods for reducing TCO [54]. The costs considered in these two categories include the necessary hardware and software for supporting system centralization or for enabling conforming to standards, training of the employees for being able to manage the centralized system and specialized training for the standard systems (which may entail higher costs) [54].

The hidden costs include "training, increased development demands, system and application maintenance, and system planning" [55] as well as downtime costs, which occur not only when failures of hardware or software happen but also when maintenance or upgrading operations take place. Downtime costs contain the costs for the nonworking systems, the nonworking employees, the repairs which again include personnel and equipment costs [54] [55]. Another factor that can be considered as hidden is the cost for measuring the cost factors [56]. Some cost drivers can be measured more effectively than others, reducing in that way the effort for acquiring the measurements. Moreover, a hidden cost lies in the time employees spend in using the system for non-working activities, e.g. for refining their work or fixing problems that cause trouble to them, which is called 'Futz factor' ([54] [55]).

For the rest of the cost categories, a variety of cost factors are identified in literature. A detailed list of cost factors can be found in APPENDIX II. The type of the cost factors depends on the technology and the context that TCO is going to be applied on. Mutschler [42] makes a further distinction of cost factors in static and dynamic:

- **Static cost factor**: represents factors that do not change along the course of time. They can be decomposed in other static cost factors. Examples of common static cost factors are given in Figure 14.
- **Dynamic cost factor**: represents costs that may change along the course of time. They can be decomposed in static and dynamic cost factors. Dynamic cost factors are also influenced by intangible cost factors, called impact factors [42], such as process complexity and employees' process knowledge.

A) Software Costs	**B)** Hardware Costs	**C)** Other potential SCF
- Operating system software - Package and licence costs - Application development tools - Security and encryption software - Networking software - Communication software - Systems management software - Database management systems - Front-end applications	**1) Front End Costs** - Monitors, printers, scanners, other periphals **2) Networking Equipment** - Cabling, routers, switching devices, encryption devices **3) Backend Costs** - Servers, mainfarmes, desktop PS units	- Costs for external consultants - Costs for external development - Costs defined by contracts - Costs for user training - Costs for training support personnel - Support costs - Compliance costs **Generally**: Budgeted (Project) Costs

Figure 14 - Overview of Static cost factors, obtained from Mutschler [42]

Formula. The formulas used for the calculations depend on the identified costs and their type. So, different types of cost factors or drivers require an investigation of the market prices or benchmarking, as performed by Martens et al. [53] for Cloud Computing services, or putting in place mechanisms for acquiring and keeping track of the data used as input for the calculations. Since TCO is the total cost, the final outcome is the sum of all the costs identified.

Furthermore, a close relation between the TCO and the activity-based costing has been stated in literature (e.g. [56] [55]). Activity-based costing can support the effective implementation of TCO. The main idea (derived from examples from and references to other pieces of research by Ferrin and Plank [56]) is the identification of distinct activities and determination of the TCO per activity, which then will be used for calculating the overall TCO. The activity-based logic also assists in the indication of costly activities or processes which have to be managed for minimizing their individual costs. Additionally, as described by Drury [55], the cost containment process consists of the following steps: identifying cost objects, identifying and classifying total costs, and tracing and allocation of the costs to the cost objects, which can be products, customers, departments, or processes. Important factors for quantifying the costs are the price, the quantity and time, since TCO takes into account the lifetime of an element [55].

In the case of EA, the classification of costs can be further refined in order to reflect the layers where the costs appear. A proposed schema is presented in APPENDIX II.

Advantages. TCO offers a wide range of benefits which originate from the improved communication between stakeholders (customers, suppliers, decision-makers, executives) and the analysis of the lifecycle of an IT component [53]. The analysis of all different stages of the lifecycle promotes cost savings in all of them and provides a more realistic estimation of the expected costs of an IT project from the start of the project [55]. The advantages of TCO mentioned by Drury [55] are presented in Table 12.

Disadvantages. Researchers have also identified a number of drawbacks or problems regarding TCO as a measure. Among the statements discussed by Drury [55], the most important ones are the following:

1. TCO ignores some cost factors:
    a. Complexity costs, which are of high importance for IT investments.
    b. Transition costs, which are associated to migrating to a new system, technology. They can also include the one-time costs regarding the migration besides recurring or incremental costs.
2. Emphasis on cost cutting actions can lead in neglecting the fact that IT performance and user satisfaction may be reduced and in promoting the already existing underfunding of IT operations.
3. Considering only costs for the IT evaluation can be only temporary. Monitoring and managing costs are important but IT evaluation should be not limited to these actions, since great opportunities may be lost.

More disadvantages, as stated by Drury [55], are presented in Table 12.

Advantages	Disadvantages
✓ Provides justification for decisions ✓ Supports outsourcing ✓ Emphasizes cost avoidance ✓ Avoids the easy way out ✓ Prepares for life cycle changes ✓ Forces cost savings ✓ Forces identification of tracking of technology assets	✖ Costs not comprehensive ✖ Not sophisticated ✖ Negative performance effects ✖ Does not consider value ✖ Ignores productivity gains ✖ Shortsighted in perspective, since reducing IT expenses does not necessarily increase value return ✖ Difficult to accurately calculate and spread overhead costs ✖ Lack of industry standard for measurement and comparison of costs with other organizations ✖ Fantasy document due to difficulties to obtain accurate calculations

Table 12 - Advantages and Disadvantages of TCO, based on Drury [55]

3.4.4. Activity-Based Costing (ABC)

Explanation. Activity-based costing is a method for determining products, services and in general objects and allocating costs to them [42]. It has improved over the years and it has turned into a profit improvement tool which integrates a number of analytic costing methods for providing financial and organizational information [57]. ABC focuses on the work activities of people and equipment by which a product or service is realized and on what these activities consume [58]. One of its main capabilities is tracing indirect costs to products, services and customers by assigning resources, costs and performance measures to the work activities [58].

ABC provides a method for measuring the costs of resources actually used by an activity by determining the quantity of the resources needed for producing the output [59]. The *cost of the resources used* can be compared to the *cost of resources supplied*, which indicate the current expenses and the capacity to perform activities as a result of these expenses [59]. The cost of resources supplied is included in the financial statements of the organization and the difference between the two types of cost indicate the cost of *unused capacity* for a specific period [59]. Knowing the unused capacity can be beneficial for the organization as explained in the ‘Advantages’ of ABC (below).

The basic terminology used in ABC is presented in Table 13. The explanation of the terms is based on the ABC analysis by the Institute of Management Accountants [58] and by Cooper and Kaplan [59].

Term	Explanation
Resource driver	A measure of the quantity of resources consumed by an activity, for example: • the floor space occupied by the activity • time spent by employees for a work activity • energy expense in kilowatts by a machine (indirect supplied-purchased resources)
Activity driver	A factor used to assign cost from an activity to a cost object. A measure of the frequency and intensity of use of an activity by a cost object. An activity driver represents the demand that outputs place upon an activity and, hence, it is used for estimating the quantity of each activity supplied to outputs. Examples of activity drivers are: • number of sales orders processed • number of setup hours • number of receipts • number of direct labor and machine hours • number of parts maintained Activity drivers can be unit-related, batch-related, order-related, product-sustaining and customer-sustaining.
Cost object	An organizational element for which cost data is desired for planning, control and decision-making. Such organizational elements are, for example: • function • organizational department, (sub)division or work unit • contract • product • customer • process
Cost object driver	This driver applies in the case where a final cost object interacts or consumes other cost objects. The measurement of the frequency and the intensity of demands placed on an object by other objects is represented by the cost object driver.
Cost driver	A more general measure of activity which serves as a causal factor in the incurrence of a cost to an entity. Cost drivers are considered in ABC analysis for deciding on the potential impact of an event (driver) on processes or activities. They represent the 'why' in the costing process as shown in Figure 15. Examples: • direct labor hours and machine hours • computer time used • contracts

Table 13 - ABC terminology

Formula. There are two views of ABC, the *cost assignment view* which transforms the costs of resources to cost of work activities and finally to costs of the cost objects (products, services, etc.) and the process view which arranges the work activities in terms of time and the context of the business process to which they belong (Figure 15) [58].

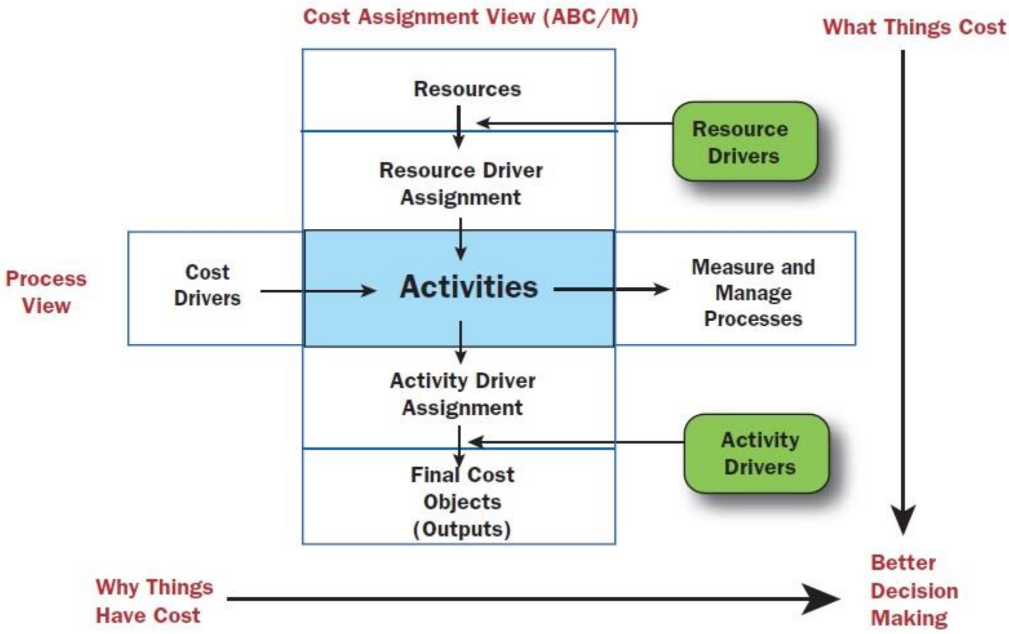


Figure 15 - ABC views, obtained from Institute of Management Accountants [58]

The activity analysis includes the determination of the activities and the resources (in terms of people, time and equipment) that are necessary for performing each activity, the total associated expenses for the necessary capacity per activity, the identification of the operational data that reflect the performance of the activities, and the delivered output of each activity based on the supplied resources. Based on these data, an estimation of the cost of supplying a unit of the outcome of the activity can be calculated. [58] [59]

In more detail, the six steps for implementing the ABC method are the following ([42] [58] [60]):

1. **Identify and define activities and activity pools**: scope of business activities under analysis and classification of activities (value adding or non-value adding, primary or secondary, required or non-required [42]). For processes that contain sequential steps, the activities that make up each step should be determined [58].
2. **Directly trace costs to activities**: determination of the various costs for each activity based on resource drivers. The costs are derived from the income statement of the company, for instance.
3. **Assign costs to activity cost pools**: calculation of total costs for each activity or activity pool by using the resource drivers identified in the previous step. The total cost assigned to the activity represents the total resources consumed by the activity or the activity pool.
4. **Calculate activity rates**: The 'activity unit cost' is calculated as a ratio of the total input cost to every activity output expressed in a measurable manner. The activity drivers are used in this step as a factor of the activity output. Activities may relate to both final cost objects and other work activities e.g. an activity may support another activity. The supportive activities can be considered as intermediate cost objects which further distribute their costs to other activities or final cost objects. Due to this fact, the total cost for an activity is calculated based on the resource drivers and the activity drivers. This approach is called multiple-stage ABC approach [58].

5. **Assign costs to cost objects:** use of each activity's driver quantity and unit activity driver cost for calculating the costs for each final cost object. Thus, the total costs are assigned to the cost objects based on their consumption of the activities.
6. **Create the *"bill of activities"* and *"bill of cost objects"*:** by assigning costs to activities and to final cost objects (e.g. customers), a direct overview of the resources consumed by them and the proportion of the overall consumption are provided. Thus, the identification of candidates for improvement is supported.

A visualization of the relationships between the activities and the cost objects is possible and it is called 'cost assignment network'. Through this network, tracing the consumption of resources and activities by the cost objects and the costs associated with the consumption are transparent [58]. The resemblance between the cost assignment network and the EA model due to the causal or dependence relationships among elements is an indicator of the usefulness of the ABC approach.

Advantages. ABC can support strategic cost management and operational cost management [58] and can provide insight for managers on how to reduce resource use, while sustaining revenues [59]. By reducing resource usage, ABC helps in identifying the created unused capacity and in how to use it as an advantage for increasing profits [59]. Furthermore, as it is shown in Figure 16, ABC is beneficial to every aspect of performance management.

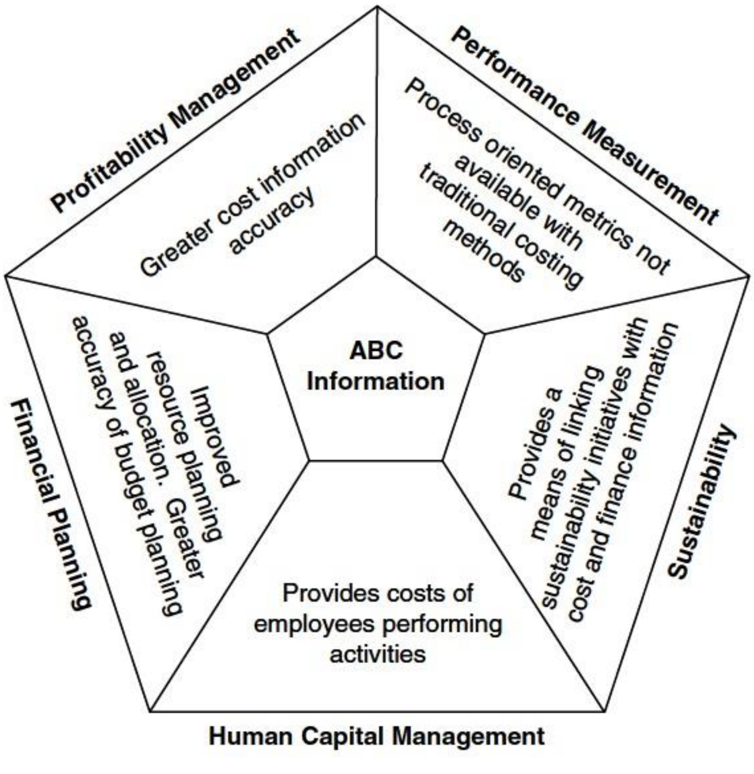


Figure 16 - ABC: The foundation of Performance Management, obtained from Turney [57]

Disadvantages. A basic problem regarding ABC comes from the misuse of the approach. It is a useful method for performance management and it should not be used as a method for calculating the cost of performing an activity in the short-run [59]. Additionally, the interpretations and the re-allocation of resources based on the analysis provided by ABC should be performed in a yearly basis, for example, and not weekly or monthly, since such a use of ABC would suggest a continuous resizing of the supplied resources. Thus, ABC can be considered as an evaluation approach which can provide insight for improvement based on data from the long-run.

3.4.5. Cost-Benefit Analysis Method (CBAM)

Explanation. The cost-benefit analysis method was developed for performing economic analysis and modeling of software systems based on their architecture [61]. The purpose of CBAM is to "model the costs and benefits of architectural design decisions and to provide a means of optimizing such decisions" [61]. CBAM can be considered as a supplementary method which elicits, documents and assigns costs, benefits and uncertainty or risks to the architectural decisions and their associated business goals and quality attributes which have been identified by a goal analysis method, such as the Architecture Tradeoff Analysis Method (ATAM) [61].

CBAM relies on architecture alternatives and supports rational decision-making regarding the design alternatives [61], but it does not offer any guidelines for developing these alternatives [62]. Additionally, it is a method that focuses on scenarios that represent changes on the architecture, while the scenarios regarding the functionality of the architecture should be already covered by the goal model analysis [62]. Thus, CBAM supports the need of evaluation of system architecture solutions, in relation to the requirements and business goals that drive the architectures, from a technical and economical perspective.

Finally, among the characteristics of architecture-based cost-benefit analysis [63] are large up-front investments, long-term capital investments, indirect expenses, potential gradual costs regarding new functionalities, financial justification of a decision based on difficult to measure or intangible costs and benefits, and application of the method on cost displacement or avoidance projects for primary and secondary activities of the value chain.

Formula. The steps of CBAM presented in this section are based on the descriptions by Kazman et al. [61] and Nord et al. [62], who related the cost-benefit analysis methodology with the ATAM for providing a complete analysis of the rationale behind decision-making regarding architecture design and supporting traceability from architecture scenarios and strategies to quality attributes to business goals. In both papers, CBAM receives as input the results of the ATAM (Figure 17), where quality attributes (QAs) refer to non-functional requirements, such as modifiability, performance or usability. Specific levels of QAs are also assigned to scenarios and can be considered as another type of goals (lower level comparing to business goals) that need to be achieved by the architecture. QAs can be compared with 'requirements' in ArchiMate® terminology. Additionally, risks and sensitivity points are used for providing confidence levels of the output of CBAM by adjusting the values of benefits and costs based on these points [62].

The CBAM consists of the following six steps [61]:

1. **Choosing Scenarios and Architectural Strategies:** Selection of high importance scenarios (already prioritized by previous analysis), QA goals levels and their associated architectural decisions. For each scenario, a set of architectural strategies is developed which incorporate the potential architecture design changes (output in Figure 17).
2. **Assessing Quality Attribute Benefits:** Correlation of benefit with degree of satisfiability of a QA goal by a strategy (eventually leading to business goals) and

relative evaluation of QA goals by assigning QA scores based on stakeholders' opinions. The QA score represents the importance of the QA goal for the architecture.

3. **Quantifying the Architectural Strategies' Benefits:** Ranking of architecture strategies (AS) based on the contribution to each QA and calculation of benefits per strategy in a qualitative manner:
$$Benefit(AS_i) = \sum_j Contribution_{ij} * QAscore_j$$
Variations among judgments of stakeholders regarding the value of each strategy for a QA goal can be used for defining uncertainty.
4. **Quantifying the Architectural Strategies' Costs and Schedule Implications:** Calculation of expected costs per architecture strategy as well as schedule implications. There is no specific suggestion on how to perform cost estimation; hence any appropriate method or measure can be used. Schedule implications can include elapsed time, shared use of resources and dependencies during implementation.
5. **Calculate Desirability:** Ranking of architecture strategies based on their costs and benefits. Kazman et al. [61] define desirability in a more qualitative manner, since they use the benefits as calculated in step 3:
$$Desirability(AS_i) = \frac{Benefit(AS_i)}{Cost(AS_i)}$$
On the other hand, Nord et al. [62] make use of the ROI, as a ratio of benefit to cost, which comprises a more quantitative measure. The benefits, though, are still calculated based on scores assigned to them by the stakeholders, which diminishes the value of the ROI measure to a qualitative one. The risks and sensitivity points can provide variations of costs and benefits and result in a range of values for the desirability of the strategy; hence, its uncertainty.
6. **Make Decisions:** Plotting the architecture strategies according to costs and benefits (x and y axis correspondingly). An uncertainty region can be also depicted per strategy. Then the comparison and selection of one or a combination of more strategies can be made.

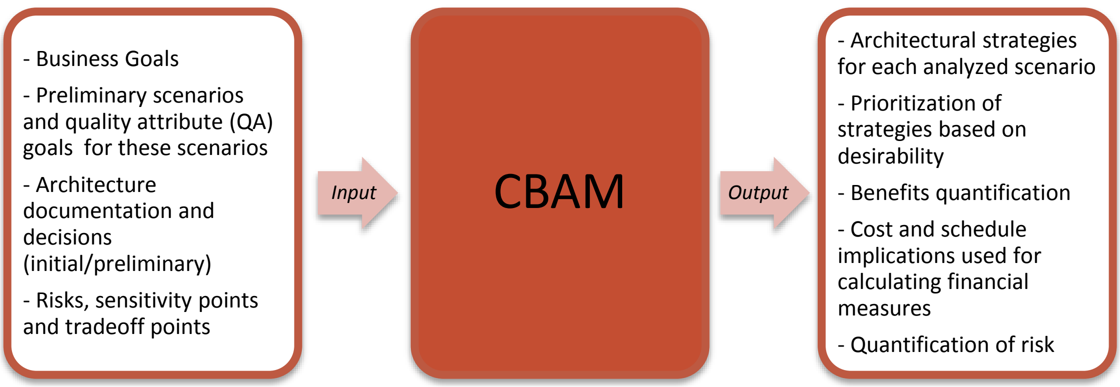


Figure 17 - Input and output of CBAM

Advantages. CBAM deals with uncertainty as it is mentioned in the explanation of the method. In particular, CBAM elicits, records and maps uncertainty onto the outcomes of the process and the decisions made though the methodology. 'Uncertainty' covers three possible forms [61]:

- Uncertainty regarding mapping architectural decisions on QA goal levels.
- Uncertainty regarding mapping architectural decisions on costs.
- Uncertainty regarding mapping QA goal levels on benefits.

Disadvantages. The first disadvantage of the cost-benefit analysis methodology entails its qualitative nature as well as its dependence of the stakeholders' opinions for defining the benefits of its strategy and quality goal. Supporting the estimation of benefits with quantitative techniques would increase the reliability of the outcomes. Secondly, CBAM may be ineffective when it is used in addition to present value measures due to the different time periods that the costs occur and the benefits are realized [63]. Finally, when CBAM is applied to information systems, attention should be paid for avoiding bias in favor of specific tasks or operations [63].

3.5. Software Cost estimation approaches

When a software system is developed or replaced, there is an amount of effort needed for building the software. In this field, a variety of approaches have been developed as well as specific metrics or factors for estimating the size, the effort and the schedule. *Estimation of size* comprises the size of the final work product; *effort estimate* is the time that is needed for producing the work product, usually in person months; and *schedule estimate* is the time in calendar months for completing the work product [64]. Various factors have been defined for estimating the size which may refer, for example, to the complexity and the dependencies of pieces of software. Describing in detail these techniques is out of scope of this thesis, but referring to some representative examples of this field provides an indication of the costs and the outcomes these techniques offer. Such approaches could be employed when a change in the architecture necessitates the development of new software and especially when the software development is done in-house.

When the effort is estimated (*number of person months*), then the cost can be estimated as [64]:

$$Cost\ estimate = \#of\ person\ months * Cost\ per\ person\ month\ (€)$$

3.5.1. COCOMO 2.0

Explanation. The Constructive Cost Model 2.0 (COCOMO 2.0) belongs to the COCOMO suite which was developed to provide reasoning about cost and schedule implications of the development process, to support investment and process improvement decisions, and to provide proof for establishing project budget and schedules [65]. Additionally, it serves trade-off analysis among "software and system life cycle costs, cycle times, functions, performance, and qualities" [66].

COCOMO 2.0 is based on the identification of scale factors and effort multipliers which serve in determining the overall effort [66] [64]. The model distinguishes among three development stages, which include different estimation procedures due to the availability and certainty of data in each stage: (1) Early Prototyping stage, (2) Early Design stage, and (3) Post-Architecture stage [66].

Formula. The general COCOMO 2.0 form of estimating effort is the following [65] [64] [66]:

$$PM = A * Size^B * \prod EM$$

where:

PM = effort in person months

A = calibration factor which reflects a global productivity average

Size = measures of functional size of a software module which can be estimated based on a 'tailorable' mix of Object Points, Function Points and Source Lines of Code. A formula exists for estimating an equivalent number of source lines of code (ESLOC) as an aggregated size estimate parameter.

B = a scale exponent which depends on five scale factors

EM = effort multipliers that influence the software development effort. There are seventeen effort multipliers which are spread over four categories of the development environment: product factors, platform factors and project factors.

Additionally, in COCOMO 2.0 the schedule estimation is provided through the following equation [66]:

$$TDEV = 3.0 * PM^{0.33+[0.2*(B-1.01)]} * \frac{SCEDPercentage}{100}$$

where:

TDEV = calendar time in months for the whole development process

PM = estimated effort in person months excluding the Required Development Schedule (SCED) effort multiplier

SCEDPercentage = the schedule compression/expansion percentage in the SCED cost driver rating.

More details regarding: (i) the exact procedures for estimating and calculating the object points, function points and source lines of code and the translation from function points to source lines of code, (ii) the scale factors, and (iii) the effort multiplier cost drivers and their ratings, which are used in COCOMO 2.0, can be found in the extended description of the method by Boehm et al. [66].

Advantages. As a part of the COCOMO suite, COCOMO 2.0 can provide a more complete and accurate cost estimation [65]. Additionally, comparing to the rest methodologies in the suite [65], it provides the bigger number of factors for calculating the estimations, which further supports the reliability of the method. Furthermore, the fact that COCOMO 2.0 takes into account the three different stages of the development of a software product and provides distinct procedures for estimation in each stage is another advantage.

Disadvantages. For the calibration factor as well as for the calculation of the scale exponent, there is a heavy reliance on historical data and on local project data. Thus, proper and careful adaptation to the data can improve the accuracy of the model.

3.5.2. Work Breakdown Structure (WBS)

Explanation. The work breakdown structure is a methodology for organizing a project by systematically analyzing it into work packages and further into detailed work activities. The refinement of the packages is performed in a hierarchical manner. The top level decomposition of the project should be deliverable-oriented, while the bottom level of the WBS consists of assignable work activities. ([67] [68] [69] [70])

WBS assists and supports project budgeting in terms of time and resources, since such information can be assigned to the bottom-level enumerated activities [67]. The fact that these activities can be directly assigned to people makes WBS a good tool for labor division as well [69]. Thus, WBS is a tool that simplifies the tasks of budget estimation and control of the project [70] and provides visibility during the development of the product [67], which makes it popular in the field of project management. Additionally, WBS focuses on the 'what' is to be done and 'what' is to be produced [68].

Variations of the WBS exist which again promote the hierarchical structure for supporting project management. An important variation is the Product Breakdown Structure (PBS). PBS is a tool for project planning used by PRINCE2® [71]. PBS "establishes a hierarchy of deliverable products required to be produced on the project" [72]. So the focus is on the deliverables and not the activities as a starting point for planning the necessary effort for completing the project [71]. The breakdown process starts with the end product which is further decomposed in sub-products, i.e. smaller deliverables [71]. The basic difference between the WBS and PBS is that PBS focuses on the deliverables, while WBS focuses on the operations and processes [72]. In order to maximize the gain from using the breakdown approaches for project planning, a combination of the methods would be ideal. WBS can become the tool for managing the work needed for each product defined in the PBS [72].

Formula. There are two types of hierarchies in WBS; the first one represents the software product itself in terms of deliverable-oriented groupings, and the second is a hierarchy of the activities needed for building the product [68] [70]. During the breakdown process, there is a point where deliverables turn into activities. These activities should be of a manageable and suitable size for monitoring, independent (to the degree possible) and unique, so that no overlap exists, and they should have quantifiable inputs, outputs and schedules [67] [68]. By meeting and following these criteria, costs, people, time and other resources can then be directly assigned to the bottom-level activities by performing the following steps [68]:

- Add duration and dependencies in order to build the logic network
- Add the calendar to the logic network to give the schedule
- Add resource names to each activity
- Add costs to each activity

By having available on the WBS all these information, the budget of the total project can be calculated as well as the cash flow by making use of the schedule [68].

Moreover, since WBS has a hierarchical structure, by assigning probabilities to the costs of the elements in the hierarchy, it is also possible to calculate the overall expected value for the total project development by rolling up from the bottom level [70].

Advantages. The WBS methodology has plenty of advantages. Firstly, it is good for planning and control of the project as well as for cost accounting and reporting of the project, since it provides a way for organizing the tasks associated with the development of the project and a method for estimating costs and schedules accurately. Secondly, WBS is an effective tool for resource management and promotes responsibility through the direct assignment of activities to people. Thirdly, WBS provides visibility and transparency throughout the whole project because of the clear overview of the project through the assignment of costs and time to each element of the hierarchy. Finally, it can be combined with other structure-oriented methodologies, such as the organizational breakdown structure, in order to increase visibility throughout the organization. ([68] [69] [70])

Disadvantages. Firstly, building the WBS depends on expertise-based recommendations. This has a major influence on the structure, since WBS relies on the assumption that the person who makes the estimate has adequate domain expertise and experience which will not lead to biased estimates [64] [70]. Secondly, WBS requires a significant amount of time and effort for building and maintaining it [69]. Finally, WBS can be considered a relatively rigid structure, which may limit flexibility regarding changes during the life cycle of the project [69] and increase scalability problems for sensitivity analyses [70].

3.6. Time-based cost estimation

3.6.1. Time-driven Activity-Based Costing

Explanation. Time-driven Activity-based Costing (TDABC) is a new version of ABC which was developed in order to deal with a few deficiencies of the ABC methodology regarding the implementation and collection of data [73]. The main concepts of the TDABC are the following:

- **Aggregated view of *resource*:** a 'resource' consists of a variety of resource types, e.g. materials, services, labor, etc.
- ***Time* as a primary cost driver:** allocation of resource cost directly to cost objects based on time, which replaces the assignment of resource costs to activities before relating activities to cost objects. 'Time' is closely related to capacity, which can denote for some resources the amount of time that a resource is available to perform work. Practical capacity, in contrast to theoretical capacity, is used as a more representative value for the actual available capacity of resources.
- ***Demand estimation* of resources per activity, activity output or cost object:** instead of first tracking costs to activities. Thus, there is need for: (i) the capacity cost rate (unit cost of supplying capacity), and (ii) unit times of resource capacity consumed by activities or cost objects. Comparing to ABC which necessitates the identification of a list of activities representing the variations among activities, TDABC employs time equations for handling variances of resource capacities in an activity. Thus, the time needed to perform a process (i.e. the activity unit time) depends on the various instances (i.e. activities) of this process and the time needed to perform each one of them:

$$T = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n$$

where:

β_0 is the standard time for performing the basic activity

β_i is the estimated time for activity i

X_i is the quantity of activity i

Formula. The steps described below are aligned with the graph presented in Figure 18 (based on Figure 2 in the paper by Szychta [73]) and they denote the basic formulas used in each part of the TDABC methodology for calculating the total costs of activities over a period [73]. Szychta [73] provides an example in her paper as well for demonstrating the difference between the traditional ABC and the TDABC. The basic steps are the following:

1. ***Determination of 'practical capacity'***: any suitable method can be used for the estimation of practical capacity. A few examples are given by Szychta [73] which include using a percentage of the theoretical capacity, taking into account activities that cause downtime or considering reserve time. Additionally, the capacity measures should be representative of the process or product resources to which they refer, e.g. personnel time, cubic meters for space capacity, etc.
2. **Capacity cost rate calculation:**

$$Capacity\ cost\ rate\ (s) = \frac{total\ costs\ of\ capacity\ supplied}{practical\ capacity} = \frac{KO}{H}$$

   This rate should be calculated separately for each product, process or service, even if the same type of resources is used.
3. **Estimation of the *amount of time required to perform each activity*** (h_i) can be achieved by observations, interviews or historical data.
4. **Activity cost driver rate calculation:**

$$Activity\ cost\ driver\ rate\ (k_i) =$$
$$activity\ unit\ time * unit\ cost\ of\ capacity\ =$$
$$h_i * s$$

5. **Total costs of activities calculation:**

$$Total\ costs\ of\ activities\ (K_i) =$$
$$activity\ cost\ driver\ rate * \#of\ activities =$$
$$k_i * x_i$$

Advantages. TDABC is used for monitoring the costs and utilization of practical capacity of entities in an organization, which is further used for measuring the cost of the unused capacity. This concept is similar to the one described in the section referring to ABC. Additionally, the capacity cost rates calculated in TDABC are based on budgeted expenses of departments or processes, which reduces the error caused by variations in actual costs [73]. Furthermore, there are practical advantages concerning the TDABC, which comprise the ease of updating the model when a change occurs, e.g. an addition of a new activity or need for updating the activity cost driver rates. The use of time equations is another advantage, since they incorporate non-standard activities in the calculation. [73]

Disadvantages. The allocation of costs of the unused capacity should be done very carefully and should be the result of careful and through analysis. Moreover, the estimations and the outputs of the model rely on expert-based estimates made by managers. Finally, the accuracy of the model can be increased by integrating it with systems providing operating data. [73]

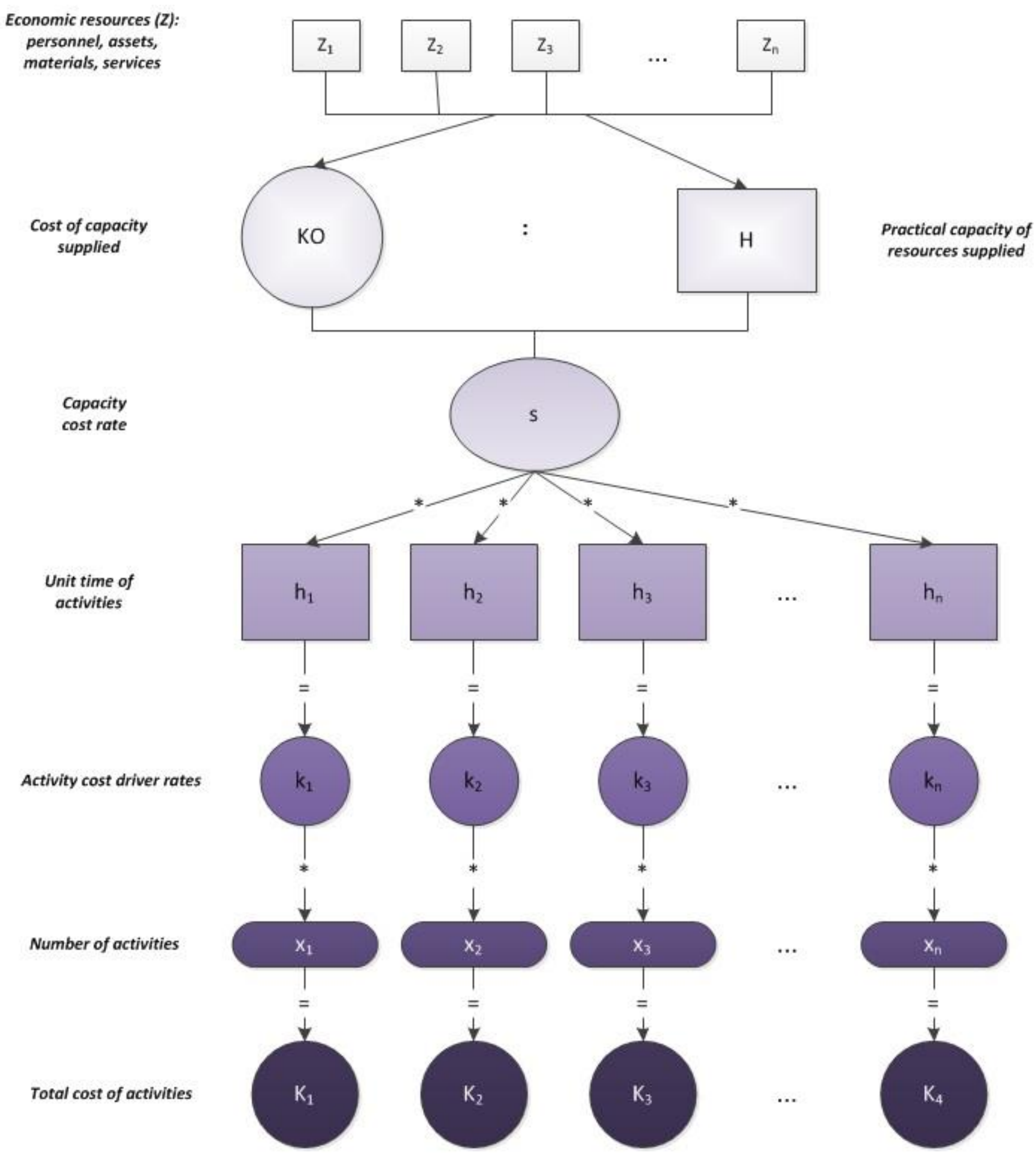


Figure 18 - Time-driven ABC

3.6.2. Times Savings Times Salary Model

Explanation. The times saving times salary model (TSTS) relies on the assumption that an employee's salary reflects his/her contribution or value to the organization [42]. Thus, the purpose of this method is to estimate the value of an IT investment in relation to the employees' work time it will save.

Formula. The calculation of the added value of the investment in the TSTS model is based on the improvement caused by replacing one process, activity, service, etc. with another one. The formula used for the measurement of the value is the following [42]:

$$Value = \frac{x_j - y_j}{T} * W$$

where:

x_j = hours per week that an employee currently devotes to an activity j

y_j = hours per week that an employee will devote to activity j after the investment

T = work time (in hours) of an employee per week

W = the wage of the employee per week in €

Thus, the calculated value represents the value in money of the time saved due to the investment.

Advantages. TSTS is a useful method for evaluating work performance and business process performance [42]. It is also rather simple to apply and it provides a direct translation of the improvement in money.

Disadvantages. The TSTS method is quite simplistic since it is based on a number of assumptions [42], e.g. saving a percentage of the employee's time equals to the percentage of the employee's costs and that this time is going to be reallocated productively in the other activities that the employee performs. Additionally, it is implied that the benefits from the IT investment are realized immediately and, hence, it does not take into account the time perspective as well as the possibility of the saved time not leading in the reduction of costs.

3.7. Performance analysis

3.7.1. System Performance Measures

Performance analysis is an important aspect in the analysis process, since it includes the quality attributes as described in the quantitative analysis process in Section 3.1. The focus in this section is on the quantifiable quality measures, which can be of two types [74]:

- Performance measures
- Reliability measures

Furthermore, Brandon-Jones and Slack [52] discuss a variety of techniques for quantitative analysis which include quality measures as well. Quantitative techniques related to quality measurement can be descriptive or evaluative. While descriptive techniques, such as the ones presented in Table 14, are used for describing the situation, measuring what is happening and, hence, for increasing the understanding of the observed situation.

The evaluative techniques, on the other hand, aim to provide support for decision-making by comparing alternatives. Among the evaluative techniques, *sequencing* deals with the decision-making regarding planning and one of its objectives is to maximize completion time or facility utilization [52]. Thus, they can be used as supplementary techniques for the evaluation of the descriptive measures. The most popular sequencing rules are the following:

- FIFO (first in, first out) and LIFO (last in, first out)
- LOT (longest operation time) and SOT (shortest operation time)

The performance or efficiency measures and the reliability measures are presented in Table 14.

Measure	Description	Formula	EA layer
Performance/Efficiency measures			
Workload	This measure represents the demand or overhead imposed by the applications, services or processes the end-users use. This demand determines the effort required from the resources and the behavioral elements (services, functions, processes) in order to provide the service to the end-user. The effort can be further represented in terms of performance measures or costs. [38] [2]	The workload is captured by an abstract stochastic arrival process and it is calculated as the arrival rate (λ) of requests to a node. The arrival rate is determined by adding the requests from higher layers to the local arrival frequency in an EA model. [38] [2] $$\lambda_a = f_a + \sum_{i=1}^{d_a^+} n_{a,k_i} \lambda_{k_i}$$ where: f_a : arrival frequency *n :* average number of uses/accesses	• Business layer • Application layer • Infrastructure layer
Throughput	Throughput is the number of transactions/requests that are completed per time unit ([38], [2], [75]). It is equal to the *arrival rate* when there are no overloaded resources ([38], [2]).	$$\boldsymbol{throughput} = \frac{\#\ of\ requests}{T}$$ *T* : observation period There is also a mathematical expression, called Little's law, for calculating the throughput time of an activity or process. It is a relationship between throughput, work-in-progress and cycle time [52]: $$\boldsymbol{Throughput\ time} = Work\_in\_progress * Cycle\ time$$ *work_in_progress* : demand in requests or transactions *cycle time*: can be calculate as the $$Cycle\ time = \frac{process\ time}{\#\ requests/transactions\ \ processed}$$ It can be used for calculating the throughput efficiency, which indicates the difference between processing time and *work content*, which is the total cycle time. [52] $$\boldsymbol{Throughput\ efficiency} = \frac{work\ content}{throughput\ time}$$	• Business layer • Application layer • Infrastructure layer

Processing time	The processing time is the time of the actual work performed for the realization of a certain result, i.e. it is the response time without the waiting time ([38], [2]). Additionally, processing time can include as a term [76]: • start and completion times of processes • average duration to complete a process • waiting and idle times of processes	The formula proposed in [38], [2] for the calculation of the processing time (*T*) of an internal behavioral element in an EA model, is the following: $$T_a = S_a + \sum_{i=1}^{d_{\bar{a}}} n_{k_{i,a}} R_{k_i}$$ where *R :* response time *S* : service time, which is the time spent internally for the realization of a service, without the waiting time for supporting services. It is inherited by the internal behavioral element which realizes the service.	• Business layer • Application layer • Infrastructure layer
Waiting time ratio	Waiting time refers to the proportion of the time the users wait for the system to respond [75].	According to [75]: $$waiting\_time\_ratio = \frac{waiting\ time}{task\ time}$$ *task time* can be the processing time	Business layer (since it refers to users)
Timeliness	Timeliness measures whether a process, service or activity was performed on time. It can include measures such as: delay, response time, jitter, latency [74].	Timeliness can be measured by the use of a variety of metrics. In [77], it is defined as: $$Timeliness = 1 - \frac{expected_{time} - actual_{time}\ of\ activities}{total\ expected_{time}}$$	• Business layer • Application layer • Infrastructure layer
Response time	Response time is the time between placing a request and receiving the result ([38], [2]). It can be calculated by adding the processing time and the waiting times (synchronization losses). Put in another way, it is the time to complete one specific task ([38], [2], [75]).	❖ The response time in [38] and [2] is expressed as the average response time of an M/M/1 queue. $$R_a = F(a, r_a) = \frac{T_a}{1 - U_{r_a}}$$ *where:* r_a *: a resource element* T_a *: processing time* U_{ra} *: utilization of the resource* ❖ Additionally in an M/M/1 queue [52]: • Utilization:	• Business layer • Application layer • Infrastructure layer

		$$\boldsymbol{u} = \frac{arrival\ rate}{processing\ rate} = arrival\_rate * mean\_processing\_time$$ • Expected waiting time in the queue: $$\boldsymbol{W_q} = \frac{u}{1-u} * mean\_processing\_time$$ • Expected work in progress in a queue: $$\boldsymbol{WIP_q} = \frac{u^2}{1-u}$$ By taking into consideration these equations and the fact that: $$\boldsymbol{Response\ time} = processing\ time + waiting\ time$$ $$Response\ time = t_e + \frac{u}{1-u} * t_e$$ $$Response\ time = \frac{t_e}{1-u}$$ where: t_e : the mean processing time	
Completion time	It is the time for completing one instance of a process (instead of a specific request). In a horizontal fashion, it could be calculated for measuring the performance of business processes. [38] [2]	Sum of processing times or response times	• Business layer • Application layer • Infrastructure layer
Utilization	It is the percentage of the operational time that a resource is busy. Utilization measures the effectiveness of using a resource and it can function as an indication of a potential resource bottleneck. [38] [2] As resources in an EA model can be considered, for example, a business actor, an application component or a device ([38] [2]). As a more general definition, resources are the elements needed for the execution of a process or an activity as well as input and	The formula given in [38] and [2] is: $$U_r = \frac{1}{C_r}\sum_{i=1}^{d_r} \lambda_{k_i}\ T_{k_i}$$ where: *d* : the number of elements that the resource is assigned to *C* : the capacity of the resource More generally put [52]: $$U = \frac{actual\ output\ of\ the\ process}{capacity}$$	• Business layer • Application layer • Infrastructure layer

	output data necessary for the execution of the process [76]. An important factor for the utilization of a resource is the size of the work queue of the resource [76].	This measure can also refer to efficiency, in more general terms, by using the following formula [52]: $$Efficiency = \frac{actual\ output}{effective\ or\ actual\ capacity}$$	
Reliability Measures			
Availability	A process, service or activity is considered unavailable when it has failed or when it is being repaired, either because of failure or because of a preventive maintenance [52]. Therefore, under the term 'availability', the following two parameters can be considered: • mean time between failures (MTBF) [74][52] • percentage of time available [74]	The mean time between failures can be calculated as [52]: $$MTBF = \frac{1}{\#\ of\ failures}$$ While the availability is [52]: $$Availability = \frac{MTBF}{MTBF + MTTR}$$ where: $MTTR$: mean time to repair	• Business layer • Application layer • Infrastructure layer
Mean downtime	It is the average time that a process, service or system is unavailable when a failure occurs before it starts working properly again [75]. It can be considered the mean time to repair (MTTR), as downtime also occurs when a preventive maintenance takes place.	It is calculated as [75]: $$Mean\ downtime = \frac{total\ downtime}{\#\ of\ times\ a\ compoenent\ is\ out\ of\ service}$$	• Business layer • Application layer • Infrastructure layer
Error rate	Error rate [74] or failure rate [52] is the measure of failures over a period of time.	Failure rate can be calculated as a percentage or as a number of failures over time [52]: $$Failure\ rate^{\%} = \frac{\#\ of\ failures}{total\ \#\ of\ tested\ objects} * 100\%$$ $$Failure\ rate^{time} = \frac{\#\ of\ failures}{operating\ time}$$	• Business layer • Application layer • Infrastructure layer

Table 14 - Performance and Reliability Measures

3.7.2. Enterprise Performance Ratios

As EA is a representation of the systems and the processes of the organization, the determination of ratios, which will indicate the performance of the organization based on information derived from the architecture, would be a useful and valuable way for demonstrating the contribution of the systems to the environment. Thus, existence of performance ratios that represent the inside-out and outside-in views of the EA are essential [78]. Hence, enterprise-oriented performance ratios can be considered as part of the outside-in view. The following are examples of structural performance ratios, as proposed by Potts [78]:

- *structural productivity of the enterprise*: defined as the return (profit) produced from each year's investment in operating expenses
- *operating income per unit of staff expense*
- *profit per transaction*
- *revenue per unit of operating expense*
- *operating expense per unit of revenue*

3.8. EA-specific Analysis Techniques

3.8.1. Performance and Cost Analysis of EA

Explanation. An analysis approach of quality aspects of EA including both performance and costs generated throughout the architecture has been proposed by Jonkers and Iacob [38]. The performance and cost analysis methodology transforms the design model provided by ArchiMate® into an analysis model through a normalization step and applies the analysis on the normalized model. The performance analysis of EA has also been described by Iacob and Jonkers [2] and Lankhorst [31].

The basic idea of the approach relies on the hierarchy of layers of EA, which are refined in service layers, i.e. exposing the functionality used by other layers, and realization layers, i.e. where the complex implementation of the functionality takes place ([2], [38]). In the realization layer three types of elements can be found: (1) internal behavior elements, such as processes, functions and system software, (2) objects accessed by the behavior elements, and (3) resources assigned to the behavior elements. There is a simplification regarding the resource elements, since business actors, application components and devices are all grouped under the 'resource' label [38].

The analysis method receives as input the data presented in Table 15 and provides as output the performance and cost measures presented in Table 16 ([2], [38]).

Elements or Relationships	Related input data
Relationships (e) 1. 'used-by' and 'access' 2. 'realization' and 'assignment'	1. n_e = average number of uses/accesses 2. $n_e = 1$
Concepts 1. Any behavior element (a) 2. Service concept	*Performance* 1. S_a = service time, i.e. time spent internally for the realization of the service f_a = arrival frequency (Usually, f is specified in the top layer of the architecture model.) 2. S of a service equals S_a of the concept that realizes it (inherited) Thus, S needs to be specified for either one of the nodes. *Costs* k_a = fixed cost per execution of a behavioral element
Concepts Any resource (r)	*Performance* C_r = capacity of a resource (by default is set to 1) *Costs* k_r = fixed cost per use of a resource v_r = variable cost of a resource = cost per time unit/tariff
Concepts Any object (o)	*Costs* k_o = fixed cost per access for an object

Table 15 - Input data of the Performance and Cost analysis approach

Element	Related output data
Any behavior element (a)	*Performance* λ_a = *workload or arrival rate* *The <u>throughput</u> for each node equals* λ *when no resources are overloaded.* T_a = *processing time* R_a = *response time* *Costs* K_a = *total cost*
Any resource (r)	U_r = *utilization of a resource*

Table 16 - Output of the Performance and Cost analysis approach

The mapping of the data and measures on the elements of the architecture is illustrated in Figure 19.

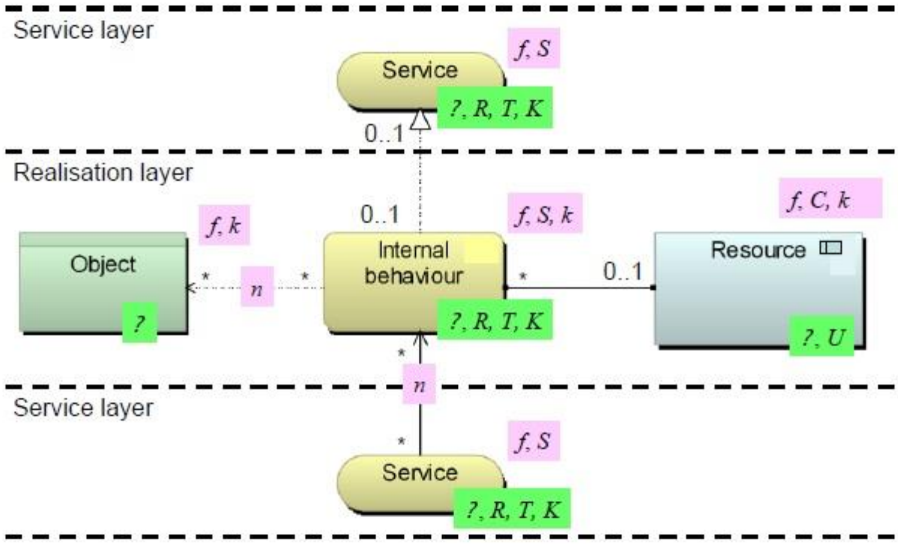


Figure 19 - Input and Output data for analysis, obtained from Jonkers and Iacob [38]

Formula. The approach consists of two phases:

- **Phase 1.** In this phase the normalization of the design model takes place ([2], [38]). An example of a transformation rule is provided in Figure 20.

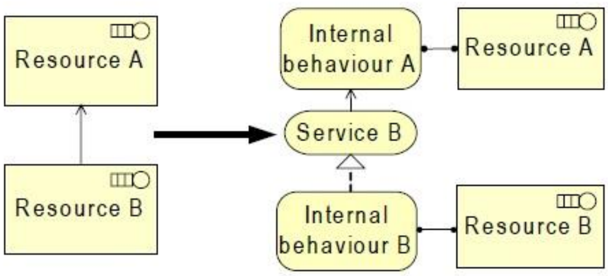


Figure 20 - Transformation rule, obtained from Jonkers and Iacob [38]

- **Phase 2.** This step consists of the performance analysis which is divided into two steps. The first one is a top-down analysis for calculating the workload (λ) of the elements and the second one is a bottom-up analysis for calculating the rest performance and cost measures (Figure 21).
  Top-down workload calculation. The *arrival rate* (λ) is calculated in a recursive manner with the following formula (used also in Table 14):

$$\lambda_a = f_a + \sum_{i=1}^{d_a^+} n_{a,k_i}\,\lambda_{k_i}$$

where:

f_a = local arrival frequency

d_a^+ = out-degree of node *a*

k_i = a child of *a*

Bottom-up performance and cost calculation. In this step, the rest of the performance measures (used also in Table 14) as well as the cost measures of all the elements are being calculated. The formulas are the following:

Bottom-up calculated measures	Description and Formulas
Utilization	$$U_r = \frac{1}{C_r}\sum_{i=1}^{d_r} \lambda_{k_i} T_{k_i}$$ where: d_r : the number of elements that the resource is assigned to k_i = an internal behavior element C : the capacity of the resource
Processing time	$$T_a = S_a + \sum_{i=1}^{d_a^-} n_{k_{i,a}} R_{k_i}$$ where: d_a^- = in-degree of node a k_i = a parent of a
Response time	Expressed in terms of a function that depends on the a and r_a. Thus, the response time depends on the way the nodes are modeled. The end formula is used for M/M/1 queues. $$R_a = F(a, r_a) = \frac{T_a}{1 - U_{r_a}}$$ *where:* r_a *: the resource element assigned to a* T_a *: processing time* U_{ra} *: utilization of the resource*
Costs	***Static costs*** Total costs of a behavior element a $$K_a = k_a + k_{r_a} + Sv_{r_a}\sum_{i=1}^{d_a^-} n_{s_i,a} K_{s_i} + \sum_{j=1}^{d_a^+} n_{a,o_j} k_{o_j}$$ where: d_a^- = number of incoming 'used-by' relationships s_i = a service used by a d_a^+ = number of outgoing 'access' relationships o_j = object accessed by a r_a = resource assigned to a Total costs of a service s $$K_s = K_b$$ where b is the internal behavior element that realizes s ***Variable costs*** Internal behavior element or service: $$V_a = \lambda_a * K_a$$ Resource or object: $$V_a = \lambda_a * k_a$$

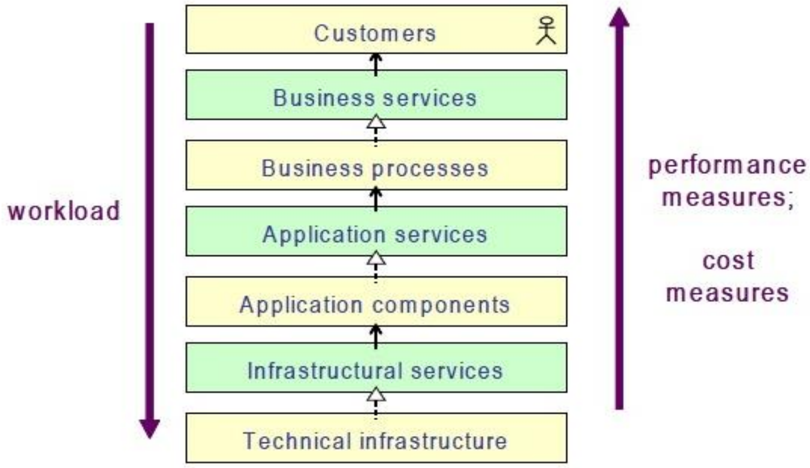


Figure 21 - ArchiMate® layers and relationships used in the analysis, obtained from Jonkers and Iacob [38]

For identifying the most appropriate performance and cost measures when an analysis of an EA model is decided, the defined viewpoints proposed by Iacob and Jonkers ([2], [38]) can be used as guidelines. The viewpoints as well as the corresponding cost and performance measures are presented in Table 17.

Viewpoints	Cost Measures	Performance Measures	Stakeholders
User/customer view	Cost per use of service	Response time	- Customer - User of an application/system
Process view	Cost per completion of process	Completion time	- Process owner - Operational manager
Product view	Cost per completion of product	Processing time	- Product manager - Operational manager
System view	Cost per time unit of using the system	Throughput	- System owner - System manager
Resource view	Resource cost (cost per unit of using the resource)	Utilization	- Resource manager - Capacity planner

Table 17 - Viewpoints, Cost & Performance Measures and Stakeholders

Advantages. This methodology provides a unified way of evaluating an EA model from the performance as well as the cost perspective. Additionally, it is a method which offers an adjustable and extendable way of working for different performance measures. The performance and cost analysis approach can also be combined with other analysis techniques in a horizontal level, e.g. techniques for process completion times, in order to provide a more detailed outcome.

Disadvantages. The main issue does not refer to the method itself, but to the lack of implementation of the method. The approach has not been implemented by a tool yet as well as not been integrated with the architectural design and analysis process.

3.8.2. Cost-Benefit EA Analysis Approach

Explanation. A cost-benefit analysis approach with a focus on EA was introduced by Abediniyan et al. [79]. This approach adopts the logic of CBAM and adapts it in the context of EA. The main goal of the authors is to introduce a method for EA analysis which focuses on cost and profit based on quality attributes. A part of the method is also the determination of scenarios which are assessed in terms of quality attributes for addressing organizations' utilities.

Formula. The method consists of six steps, presented in Table 18.

Steps	Description
Step 1 **Determination of scenarios of the organization**	This step includes the identification of the change scenarios regarding the organization as derived by enterprise goals. The determination should be handled by managers and consultants of the organization.
Step 1a **Determination of scenarios activities**	Decomposition of activities that realize the scenarios and determination of the executable programs to be performed are the essence of this step.
Step 2 **Determination of quality attributes**	The most important quality attributes and indicators regarding the organization are gathered through questionnaires answered by experts. The Delphi technique is also discussed as an option. Examples: flexibility, security, efficiency of services, customer satisfaction, investment absorption.
Step 3 **Determination of the levels of activities response**	For each of the activities identified in Step 1a, the level of quality attributes responses is determined by a group of experts. The level of responses is expressed as: • Best (%): highest level of achievement (reference point) • Worst (%): minimum reference point of expectations • Current (%): ratio regarding best and worst • Desired (%): ratio regarding best and worst
Step 4 **Assignment of activities benefits**	A utility is assigned to each level of quality attributes response. Experts are involved in this step as well. As utility, the authors define the profit gained from the stakeholders' viewpoint. The 100-scale is used.
Step 5 **Computation of the quality attributes utility and estimation of total cost**	The assignment of utility to each quality attribute is based on the outputs of steps 3 and 4. The authors propose the use of a divided difference interpolation method for defining the desired utility of a quality attribute. The *utility of the attribute* is the subtraction of the *desired utility* and the *current utility*. When referring to a project, the overall utility is the sum of the utility of every activity. An estimation of the project cost should also take place. No specific method is proposed, since the cost estimation depends on the type of the organization and the activities involved [79].
Step 6 **Analysis and ranking based on ROI**	ROI is calculated for each scenario identified in Step 1. ROI is defined as the ratio of benefits (sum of utilities of every quality attribute) and costs for each scenario.

Table 18 - Cost-Benefit EA Analysis Approach: Steps

Advantages. The Cost-Benefit EA analysis approach is an attempt to provide a framework for performing a cost and benefit analysis of EA scenarios. This analysis can be further used for evaluating the alternatives. Additionally, there is an implied relationship with the goal level, since the scenarios are determined based on the organization's goals. The fact that quality attributes are the primary source of benefits is another advantage of the method, since they represent the most important values of the organization, drivers for the calculations and estimations, and provide an approach for modeling benefits in a more systematic manner.

Disadvantages. There are two main disadvantages of the approach. Firstly, it is heavily dependent on experts for defining the quality attributes as well as the response levels. Of course, experts' opinion is necessary but it is also subjective. Decisions or estimations do not rely on quantified values. The approach also makes the assumption that the experts are experienced enough for determining the levels and utilities. The last two observations lead to the second disadvantage, which is the qualitative nature of the approach. Thus, while CBAM has been utilized as a basis for developing the Cost-Benefit EA analysis approach, its qualitative nature has not been replaced by measurement techniques and concrete calculations.

3.8.3. A practical analysis technique for Cost Allocation across EA models

Explanation. This approach has not been published, but it is used internally in BiZZdesign and it has been applied in various projects of the company. It comprises a practical technique for propagating and aggregating values across EA models. The technique's primary use is focused on cost allocation. Furthermore, the approach is implemented through the Architect tool.

Formula. The cost allocation technique is, in principle, a bottom-up approach which aggregates costs in the upper-levels of the architecture by propagating the input values of lower-level concepts' attributes. The direction of the relations connecting the concepts is taken into account while performing the calculations. So far, the method utilizes simple mathematical formulas based on sums and deductions. In addition, to provide more expressiveness and accuracy in the way the costs of lower-level concepts being assigned in higher-level ones, weights have been introduced as attributes of the relations. While, usually, no differentiations are observed in the way relations are treated for the propagation of values, the user can customize the method according to specific needs.

The logic for allocating the costs across the model is based on properties or characteristics of the system or the case being analyzed. As an example, the utilization of resources or specific concepts, e.g. servers, can facilitate the assignment of weights across the EA model. The aggregated or allocated costs in the end of the method's application can be assigned in any type of EA core concept. This method has also been applied for allocating costs to work packages and plateaus of the ArchiMate® Implementation and Migration Extension. An example of the method is provided in Figure 22.

Advantages. The main advantages of this method are its simplicity and generalizability, since the method can be adjusted to different cases and settings. Its implementation in the Architect tool is also important, since, through scripting, the method can be automated, calculation errors can be reduced and the customization of the method can be enhanced.

Disadvantages. The cost allocation method across EA models is a generic method which is helpful when it is customized and applied properly. Additionally, for the method's outcome to be meaningful, there is a high dependency on the input data and on the level of expertise and experience of the people who apply the method. Finally, the derivation of the necessary input data can be difficult or time-consuming, since information regarding utilization of resources, for instance, may not be easy to find in an organization.

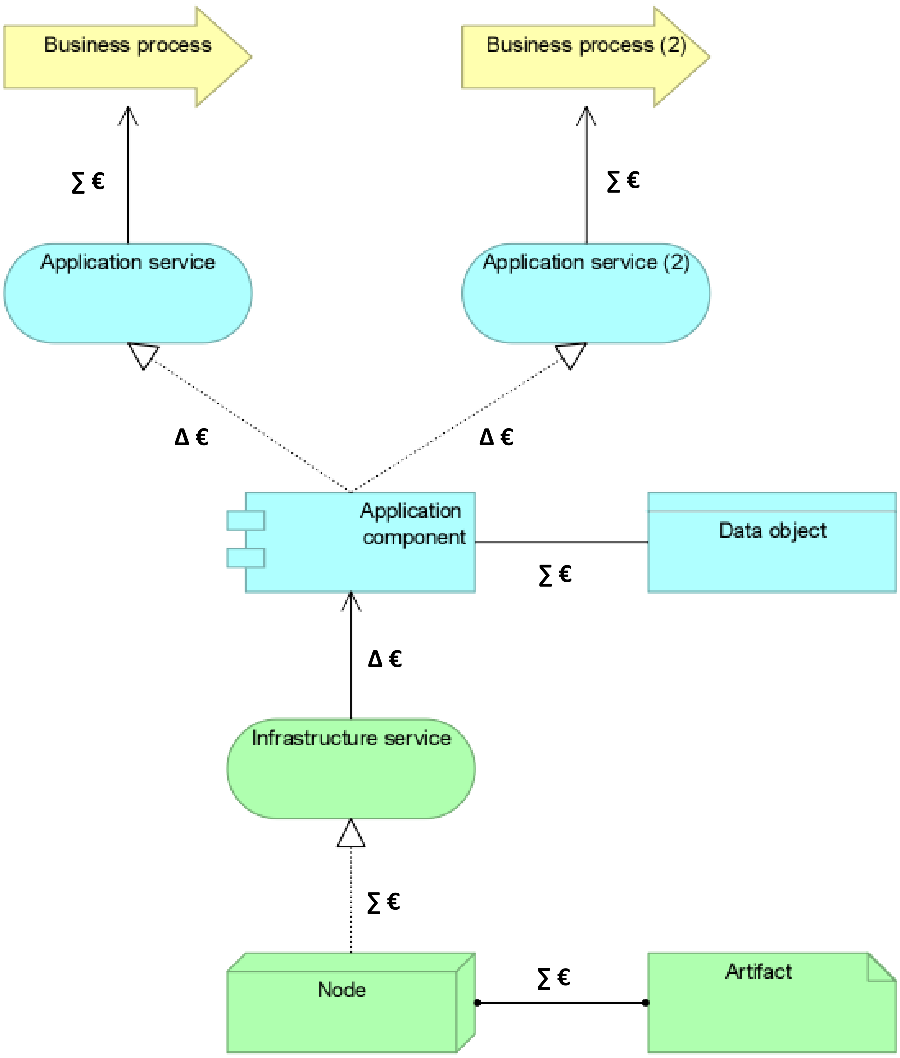


Figure 22 - 'Cost allocation across EA models' approach example

3.9. Summary

To conclude, the following figures present an overview of the techniques discussed in this chapter. Figure 23 contains all the financial analysis techniques, while the last block 'EA-specific Analysis Techniques' includes the "Performance & Cost EA analysis" method which covers performance analysis as well.

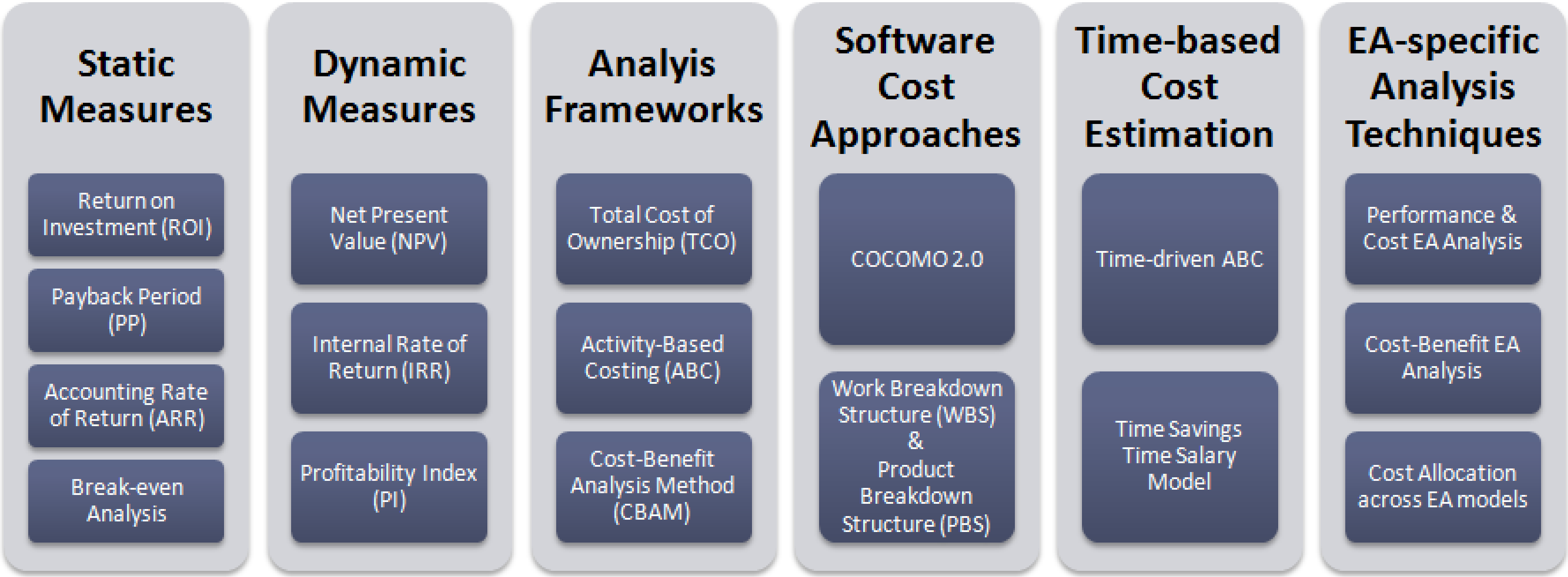


Figure 23 - Financial Analysis Techniques Overview

Figure 24 summarizes the performance and reliability measures of Table 14 and, finally, Figure 25 groups the benefit analysis methods discussed in this chapter.

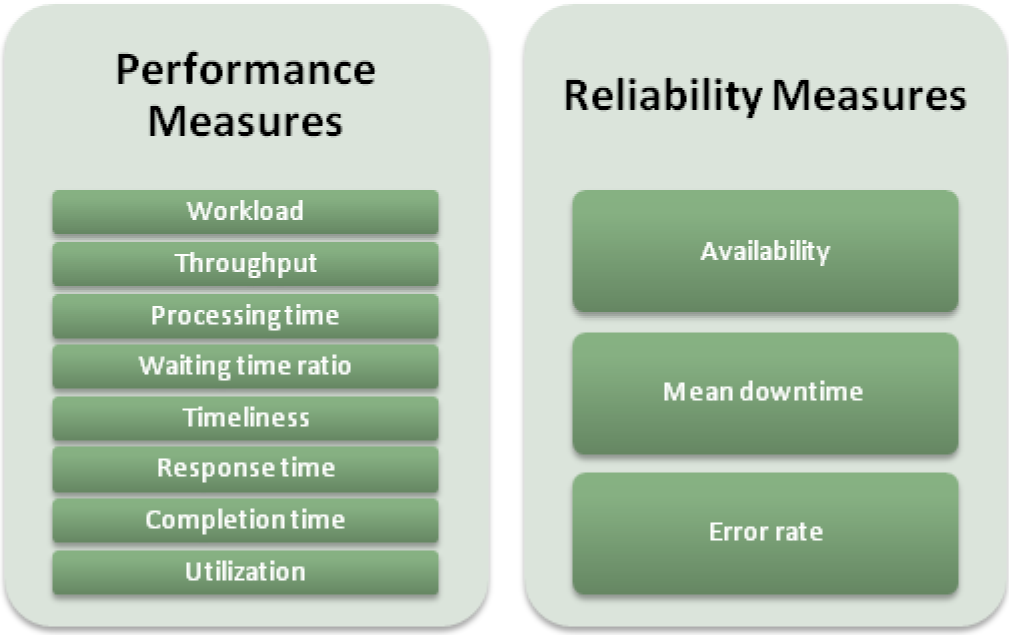


Figure 24 - Performance and Reliability Measures Overview

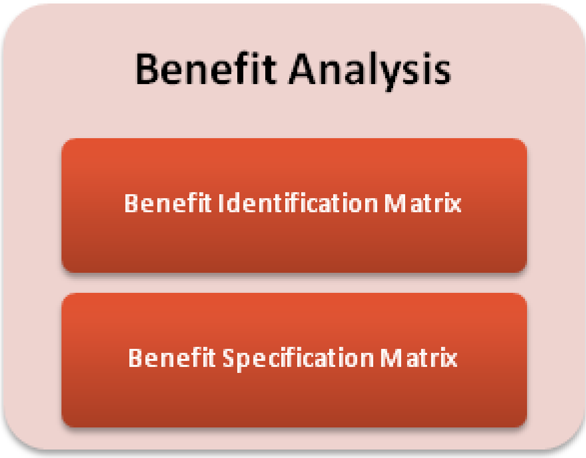


Figure 25 - Benefit Analysis

CHAPTER 4

4. Goal Model Analysis Techniques

In this chapter, the goal analysis techniques found in literature are presented. The focus is upon techniques that attempt to measure goals, assign values to relationships between goals or requirements, analyze and assess goals with respect to (architecture) scenarios and relate them with measurable attributes. Comparisons among techniques that belong in the same 'family' of analysis approach are also provided. These techniques comprise an essential element for designing the quantification mechanism(s) in this thesis, as they provide knowledge regarding the available treatments in the second step of the Design Cycle, i.e. the Treatment design.

4.1. GQM and Variations

4.1.1. Goal Question Metric Approach

The Goal Question Metric (GQM) Approach is a top-down method which aims to assist the measurement mechanism in a project by taking into account the goals of the organization and the project, and deriving (from the goals) the information that need to be assessed [80]. Thus, the GQM's goal is to operationalize goals and identify the frameworks for interpreting the gathered data [80]. Defining the way of interpreting the data is a beneficial outcome of GQM, since the method serves in providing the 'why' about the data collected and used in the measurements [81]. Identifying the reason behind the measurements also supports their reuse in future projects and activities [81].

In the GQM paradigm, as Park et al. [81] call it, measurement plays a fundamental role. According to Basili et al. [80], measurement is a mechanism for creating corporate memory, supports project planning and assists in the assessment of a project's progress and the evaluation of its outcomes. The measurement mechanism consists of a process where "numbers or symbols are assigned to attributes of entities in the real world" [81]. Three elements are essential for the measurement to take place:

- **Entities.** They are the objects of interest. Examples of entities are products, processes, resources, activities and artifacts [80] [81].
- **Attributes.** They comprise the characteristics or properties of the entities [81].
- **Rules and scales.** The rules are necessary for assigning values to the attributes, i.e. for performing the measurement. The scales used for each type of attribute and the values assigned to it are derived from the rules as well [81].

Thus, there is a need for defining the entities and attributes assessed through the measurement framework in a clear and precise way as well as the rules for performing the assignment of values. This process is facilitated by the GQM approach through the formulation of a three-level model:

1. Conceptual Level – *Goal*. The defined goals refer to the entities of the measurement mechanism. A goal is defined from various points of view and it is relative to the context it is applied on [80]. The term 'goal' does not refer to business goals, but to measurement goals [81]. Thus, there is a need for refining the high-level business goals to measurement goals before applying the GQM method. Providing clear definitions of goals is a basic step for developing the GQM model. Thus, the goal formulation consists of five parts, as it is mentioned in a more recent publication by Basili et al. [82]. The description of the five parts along with examples presented in Table 19 are based on the explanation provided by Basili et al. [82].

Goal	Description	Example	Derived from
1. Purpose **2. Issue or Focus**	1. characterize, evaluate, predict, motivate, improve 2. cost, correctness, changes, reliability, user friendliness, etc.	1. Improve 2. the performance of	the policy and the strategy of the organization [80].
3. Object (entity)	3. products, processes, resources, activities and artifacts	3. the payroll process	the description of the entities or elements of the organization that are relevant to the measurement [80].
4. Viewpoint	4. User, customer, manager, developer, organization, etc.	4. from the project manager's viewpoint	the model of the organization [80].
5. Context	5. problem factors, people factors, resource factors, process factors, etc.	5. in the context of the Financial department	

Table 19 - GQM Goal formulation

2. Operational Level – *Question*. This level consists of questions which characterize the object defined in the goal with respect to its attributes and qualities (issues). The purpose of the questions is to derive the methods for assessing the overall goal by defining the goal in a more complete manner [80]. According to Basili et al. [80], there are three groups of questions:

1. "How can we characterize the *object* with respect to the *overall goal* of the specific GQM model?"
2. "How can we characterize the *attributes* of the object that are relevant with respect to the *issue* of the specific GQM model?"
3. "How do we evaluate the *characteristics of the object* that are relevant with respect to the *issue* of the specific GQM model?"

3. Quantitative Level – *Metric*. This level includes the identification of the metrics that will be used for answering the questions of the previous level in a quantitative way. For selecting the most suitable metrics, attention should be paid on the object referring to and the models that relate to the type of the object. For the selected metrics, data must be gathered which will serve as the input for the measurement mechanism. Defining the data as well as the collection and analysis mechanisms are part of this step. The data can be subjective or objective. [80]
Various meanings and definitions have been attributed to the term 'metric' which have created some confusion in the research and practice community [81]. Buckl et al. [83] have proposed a definition of metric based on their interpretation of the GQM method by Basili et al. [80]:

> "A *metric* is at-least ordinally scaled information on a property of an object derived via an objective or a subject-dependent measurement procedure."

By applying the GQM method, a graph model is developed consisting of trees with *goals* as roots and *metrics* as leaves, while *questions* represent the intermediate layer for translating the goals into metrics (Figure 26). Overlaps may appear between answers to questions, since the same metric can be used for answering different questions. Even though a representation can be derived, the specification of the goals, questions and metrics are in a textual form and no guidelines are given for specifying the measurement process and the collection of data.

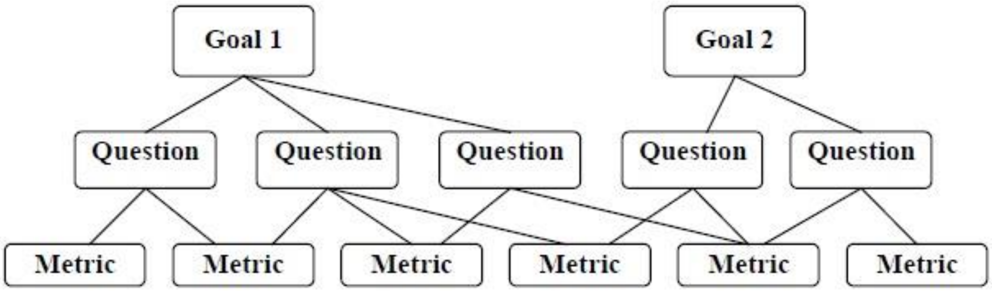


Figure 26 - GQM model, obtained from Basili et al. [80]

4.1.2. Goal-driven Measurement Process and GQ(I)M Approach

The GQ(I)M approach can be considered the sub-process of the goal-driven measurement process introduced by Park et al. [81]. The main goal of the goal-driven measurement process is to retain traceability from the high-level business goals to the measures, as a justification for the measurements being performed. For achieving this purpose, a ten-step process has been defined which starts with identifying the business goals, refining them into measurable and manageable goals and, finally, assigning appropriate measures and indicators to them.

From the ten steps that the goal-driven measurement process consists of (Table 20), the steps 5-8 are considered as the GQ(I)M method. The 'I' that distinguishes the GQ(I)M method from GQM refers to indicators. An indicator represents the expected visualization of the measurement results (originating from one or more measures) that are going to be used for answering the questions [81]. Three types of indicators (success, progress and analysis indicators) have been defined by Goethert and Siviy [84]. An indicator may be a picture, a chart (e.g. pie chart or bar chart), a diagram, a table or simple ratios depending on the type of measure they are going to display. The purpose for introducing indicators in the GQM method is for increasing the understanding of the measurement data as well as supporting the identification and definition of the right data that need to be collected by already imagining the way they are going to be displayed [81].

Another point that needs to be made regarding GQ(I)M is about the 'M' which refers to a measure and not a metric as in GQM. Park et al. [81] consider 'measure' as a more clear and well-defined term and they relate it to the widely accepted definitions of measurement. Thus, GQ(I)M stands for Goal Question Indicator Measure.

The steps of the goal-driven measurement process, as presented by Park et al. [81], are:

	Steps
Refining *Business* Goals to *Measurable* goals	1. Identify business goals
	2. Identify what needs to be known for understanding and evaluating the activities related to the business goals
	3. Identify sub-goals
	4. Identify entities and attributes related to the defined sub-goals
GQ(I)M	5. Formalize measurement *goals*
	6. Identify quantifiable *questions* and the *indicators* that will be used for achieving the measurement goals
	7. Identify the data to be collected for constructing the defined indicators
	8. Define the *measures* that will be used and make these definitions operational
Defining the measurement plan	9. Identify actions to be taken for implementing the measures
	10. Prepare a plan for implementing the measures

Table 20 - Goal-driven Measurement Process Steps: GQ(I)M

From the ten steps presented in Table 20, it can be realized that the first four steps aim to support GQ(I)M by providing a way for deriving the measurement goals from the business goals. In that way, the application of the GQ(I)M becomes more effective and the identification of the questions in Step 6 is facilitated; since the goals have turned from abstract, ambiguous and informal into concrete, measurable and manageable. The traceability from the business goals to the measures is illustrated in Figure 27.

The GQ(I)M approach is a more comprehensive approach comparing to GQM and provides a variety of templates for supporting proper definitions, descriptions and documentation of, for example, the question lists and entities in step 2, the sub-goals in step 3, entities and attributes in relation to questions in step 4, the measurement goals (more than one templates for describing different aspects of the goals) in step 5 [81]. The most intensive template is probably the one for the indicators, which, among others, includes the data elements to be used for the production of the indicator, the algorithm for combining the data, the analysis supported by the indicator and interpretation possibilities through the results [84]. The templates, on the one hand provide structure to the process, but, on the other hand, make the approach very intensive and time consuming; hence, they reduce its usability and flexibility.

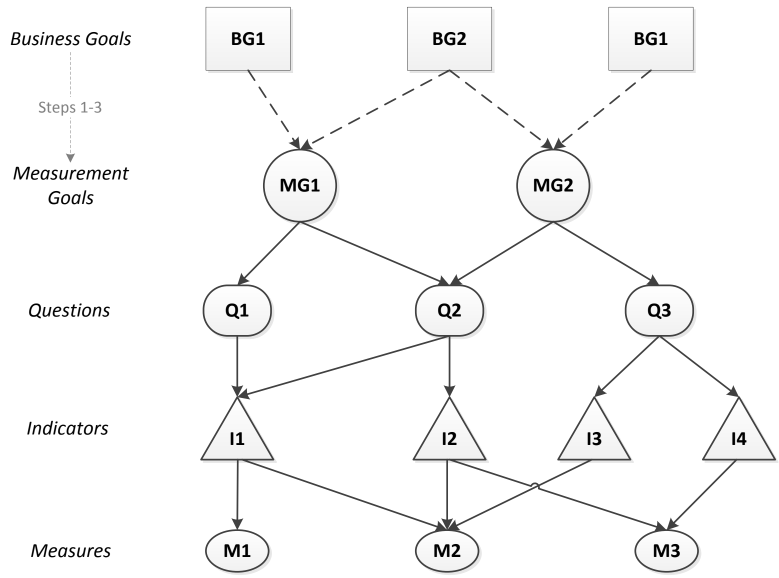


Figure 27 - GQ(I)M

4.1.3. GQM+ Strategies Approach

GQM+ Strategies [85] [86] is a framework developed for extending the functionality and the usability of the GQM model. In general, GQM+ Strategies provide a process for making the business goals and the lower-level goals more explicit through the identification of strategies. Goals are further operationalized and mapped to measures through the GQM approach. GQM+ Strategies approach introduces the concept of levels and the existence of goals in different levels, which are not pre-defined and depend on the organizational structure. In that way, GQM+ Strategies approach “creates mappings between the data related to goals at different levels, so that insights gained relative to a goal at one level can still feed up and contribute to satisfying goals at higher levels” [87].

GQM+ Strategies consists of:

1. eight *conceptual elements*, that comprise the core of the approach. The GQM+ Strategies concepts are presented in Table 21 based on the definitions provided by Basili et al. [85] [82].
2. the *grid derivation process* which includes two parallel kinds of processes for supporting: (i) the refinement of business goals to lower-level goals through strategies, and (ii) the identification of measurement goals and measures by utilizing the GQM method. This process is described in detail by Basili et al. [86], while the steps are explained in Table 22. The grid derivation process is considered a top-down process, but it is flexible, meaning that it can be initiated at any level and it can move-up or down [86] [87] [88].
3. a *template for formalizing business goals* as well as lower-level goals, while for the measurement goals the template of the GQM approach is used. The elements that are documented through the template are: (i) the *activity* for accomplishing the goal, (ii) the *focus* of the goal, (iii) the *object* being analyzed, (iv) the *quantification* defined as the magnitude of e.g. the change, (v) the *timeframe* for achieving the magnitude, (vi) the *scope*, (vii) the *constraints* limiting the goal achievement, and (viii) the *relationships* with other goals [86] [87].

Conceptual Element	Definition
Business Goals	High-level organizational goals which facilitate the objectives of the organization
Strategy decisions	Approaches for achieving the business goals
Lower-level Goals	A set of goals inherited from higher-level goals as part of the strategy related to the higher-level goal
Strategy activities or **Scenarios**	A set of activities or steps for achieving the chosen strategy
Measurement Goals	Goals which are measurable and concrete in order to facilitate the GQM process. A GQM goal is associated with goals at all levels.
Interpretation models	Models that help interpret data gathered through measurements for the assessments of the goals
Assumptions	Estimations about conditions or events that can affect the interpretation of the data
Context Factors	Environmental variables that represent the organizational environment and affect the kind of models and data that can be used

Table 21 - Definitions of GQM+ Strategies Concepts

	Step	Explanation
Tasks for defining Goals & Strategies	**Elicit General Context & Assumptions**	Provides the motivation and the rationale for the goals to be defined.
	Define Top-Level Goals	• High-level goals and prioritization • Identification of conflicts and relations • Goal formalization (template)
	Make Strategy Decisions	Identification and selection of the strategies for accomplishing the business goal
	Define Goals	• Identification of lower-level goals derived by the defined strategies, i.e. refinement of the strategies by creating a lower-level of explicitness • Goal formalization (template)
Tasks for measuring Goals	**Define GQM Graphs**	• Identification of measurement goals for goals in different levels • Definition of questions, metrics, criteria and interpretation models • Relationships between interpretation models of the current level and the higher one

Table 22 - Grid Derivation Process Steps

The grid, which is the main output of the GQM+ Strategies approach, can be perceived as a holistic overview from the business goals towards the measurements, and vice versa, through goals from various levels as derived by the strategies. The integration of all the concepts of the approach into the grid are illustrated through an example in Figure 28. Additionally, the grid is characterized as the outcome of the elicitation process for identifying and collecting the required information for goal-based measurement planning [89]. Petersen et al. [89] have also conducted a research in industry on how to best elicit the required information for performing the GQM+ Strategies based measurements and developed an elicitation instrument. The elicitation instrument, called GQM+ S-EI, adds more rationale in the process of identifying the strategies which realize a goal by extending the GQM+ Strategies approach with two types of links between goals and strategies: (i) contribution and conflict relationships between strategies, and (ii) rationale documentation on the links. The information elicited by their instrument, from the case study Petersen et al. [89] conducted, was further evaluated in terms of accuracy and completeness.

GQM+ Strategies is a domain-independent approach which combines the 'what' (goals) of the organizational strategy with the 'how' (strategies) [88]. Additionally, GQM+ Strategies, by documenting the alignment of goals, strategies and measures, supports the identification of potential future misalignments [88]. However, several challenges have been experienced during the application of the method, among which are the difficulty of "finding suitable entry points" regarding the levels and the goals for starting the process, and the lack of a tool for visual representation of the grid, which could facilitate faster feedback and decision-making [88].

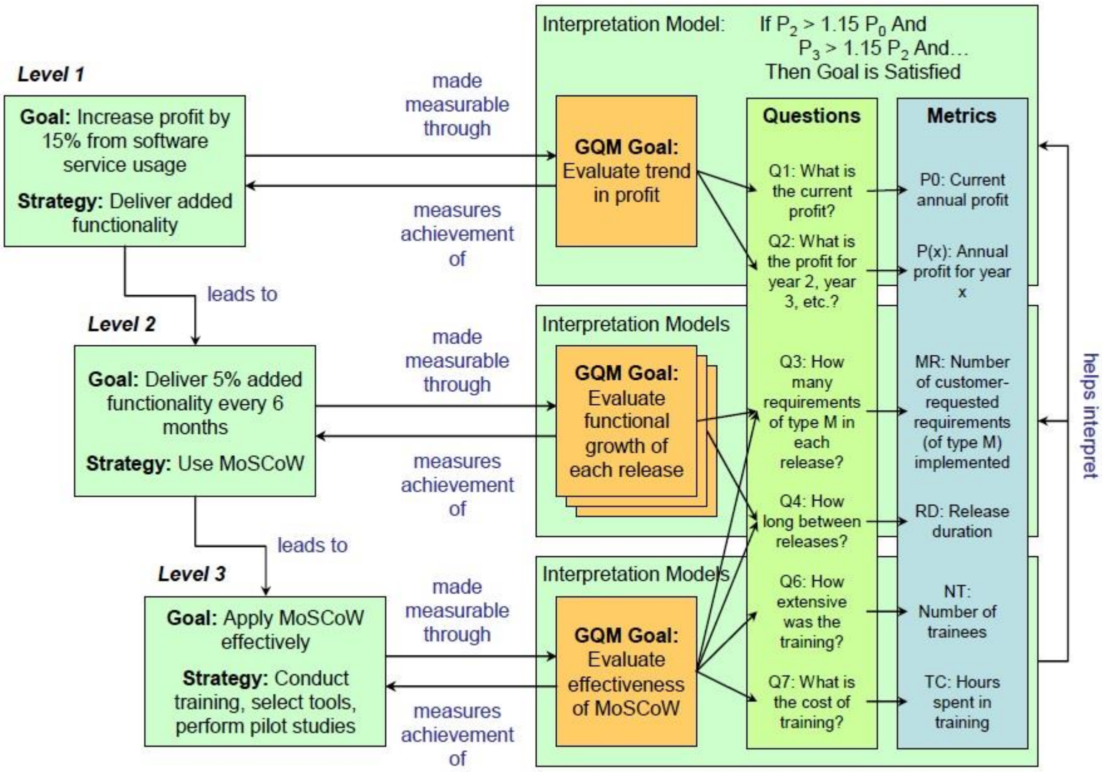


Figure 28 – Example of a GQM+ Strategies Grid, obtained from Basili et al. [82]

An attempt of integrating the GQM+ Strategies with Business Value Analysis (BVA) realized by Mandić et al. [87] is discussed in APPENDIX IV.

4.1.4. Structured Prioritized Goal Question Metrics Approach

The Structured Prioritized Goal Question Metrics (SPGQM) approach [90] was developed for providing a well-structured process for selecting measures when changes occur in the goal level in terms of priorities imposed by stakeholders, evolving goals and the dependencies among goals and measures. The SPGQM framework proposed by Tahir and Gencel [90] is an extension of GQM and it is presented in Figure 29. SPGQM consists of the following four phases:

1. **Goals Definition Phase.** In this phase the goals and their sub-goals are defined in an iterative way. The goals considered in this phase are both business goals as well as measurement goals. Both types of goals follow a similar template for structuring their definition. The primary goals are related to sub-goals through dependency relationships. The elements of the goal definition include the elements of the template provided by GQM plus a few additional ones in order to support the dependency relationships (e.g. 'relevant business goals' that relate to a measurement goal, 'information needs to track') and the traceability of the modifications that a goal may experience over time ('traceability information'). There is no specific description, though, for the refinement process of the business goals for turning them into measurement goals.
2. **Questions Definition Phase.** This phase is responsible for identifying and structuring the questions which serve the identification of the potential measures. As it is illustrated in the case study by Tahir and Gencel [90], a selection and refinement process for the questions also takes place. The structure of the questions is performed in a similar way to the goals for ensuring the coverage of the issue of the goal by the sub-issues that the question addresses and for monitoring the horizontal dependencies between questions as well as the vertical relationships between goals and questions for a question that relates to more than one goals.
3. **Measures Selection and Decision Making.** This step is responsible for selecting the measures that are going to be implemented in the measurement plan. Since the number of measures identified may be large and the collection of the data related to them may be intensive, there is a need for an optimum set of measures.

   The Optimum Measures Set Decision (OMSD) Model described by Bhatti et al. [91] is utilized for defining the optimal set of measures based on cost, resource and time constraints. The OMSD model consists of five steps.

   At some point in the OMSD model, an initial set of measures are identified based on a set of criteria defined by the company or maturity and software engineering standards (since the method was developed for the software industry). For the analysis of the measures, that meet the criteria, data related to the factors that characterize the measure are gathered to support the selection decision. The factors are illustrated in Figure 30, while formulas for calculation of the factors are also provided in the paper by Bhatti et al. [91]. The decision is based on the *effort* and the *cumulative cost* for collecting the measure. For the final decision-making, a matrix is made which relates each measure with the attributes it addresses. From the matrix the *usage* of each measure can be concluded, while the importance and the cost of the measure have been already defined. The cost and time constraints are considered and a screening process takes place for deciding on the optimum set of measures. The screening process can be considered a trade-off analysis step for selecting the most suitable set of measures when certain budget, resource and time limits as well as dependencies and priorities exist.
4. **Cost-Benefit Analysis.** The Cost-Benefit Analysis phase refers to the goals and not the measures. The OMSD is used again in this step for calculating the cost of the optimum set of measures. The cumulative cost of the measures comprises the cost of the corresponding goals. The benefits are not calculated in monetary values, but the perceived importance of the goal is examined in a qualitative way by questioning

various stakeholders. The 100 dollar method is used for assessing the final values of the goals.

The SPGQM framework is an interesting extension of the GQM model especially because of the OMSD model as well as because of its ability to keep track of the modifications that occur in the goal model. The facts that the costs for collecting the data for a measure, the necessary effort for doing so and the usage of the measure are considered, provides a more complete framework which supports decision-making not only in terms of analysis of goals regarding suitable measures but also identifying the measures that are aligned to the constraints imposed by the project. The costs for the collection of the data have been identified as hidden costs which are usually excluded or neglected according to the TCO approach.

The drawbacks of the SPGQM framework reside on the lack of a process for refining business goals to measurement goals, and on the qualitative-oriented cost-benefit analysis. However, these issues can be supported by methods already described, such as the goal-driven measurement process and GQ(I)M or the GQM+ Strategies for reaching measurement goals and financial techniques or the Benefits Specification Matrix for deriving quantitative benefits.

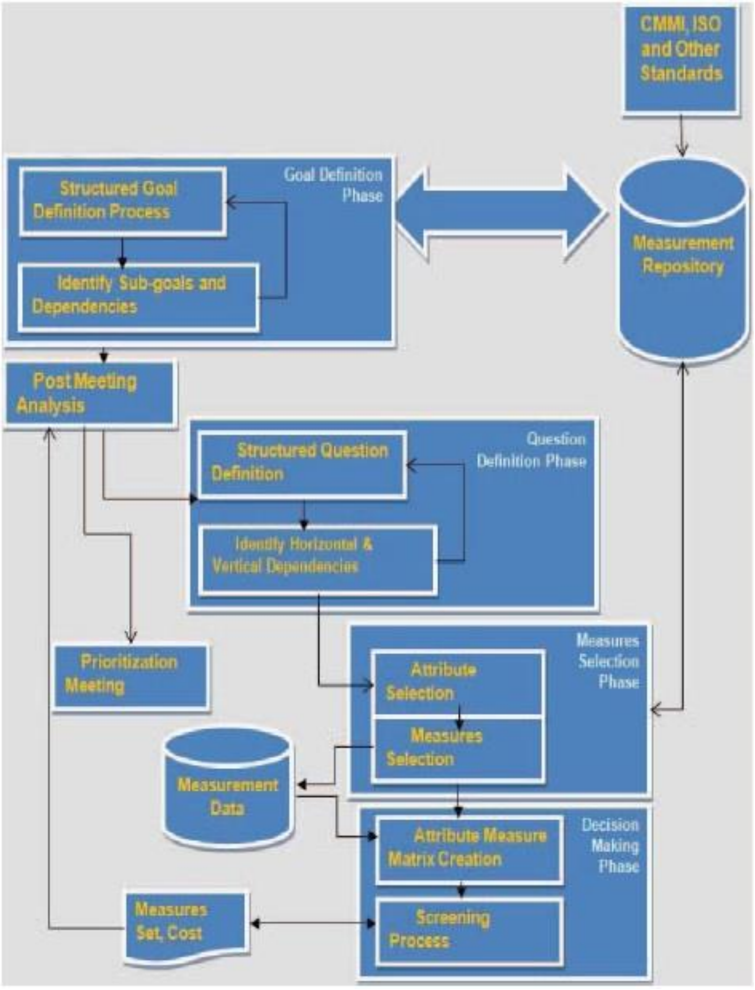


Figure 29 - SPGQM Framework, obtained from Tahir and Gencel [90]

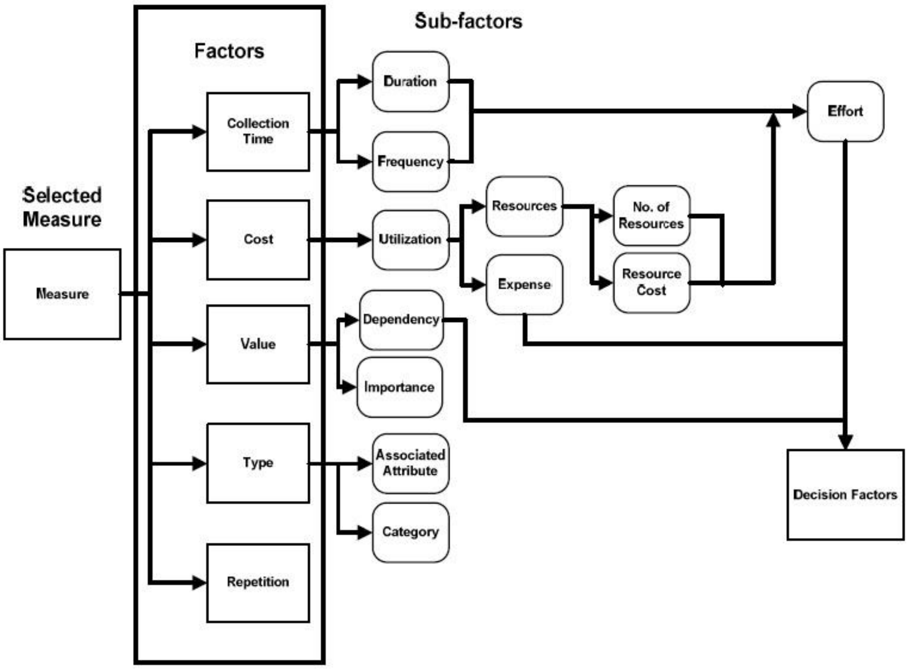


Figure 30 - OMSD Model, obtained from Bhatti et al. [91]

4.2. ISO 15939 Measurement Information Model

ISO/IEC 15939 is a standard which aims to identify the activities that are essential for identifying, defining, selecting, applying and improving software measurement in the context of a project or the organization in general [92]. The standard views measurement from two perspectives, which are combined and interrelated. The first one, includes a *software measurement process* which is responsible for matching information products to information needs in the attempt to satisfy the information needs, and the second view includes a *measurement information model* [92]. The measurement information model comprises a process for deriving and linking measures to information needs by quantifying relevant attributes and in turn indicators [92].

The ISO 15939 Measurement Information Model is further analyzed by Abran et al. [37]. In this analysis, the authors reflect metrology concepts on the process prescribed by the model for deriving measures. They also reflect quantitative analysis for the quantification of relationships across entities and attributes for resulting in the information products [37]. The ISO 15939 Measurement Information Model and the separation identified by Abran et al. [37] are depicted in Figure 31, where the basic concepts of the method are also illustrated.

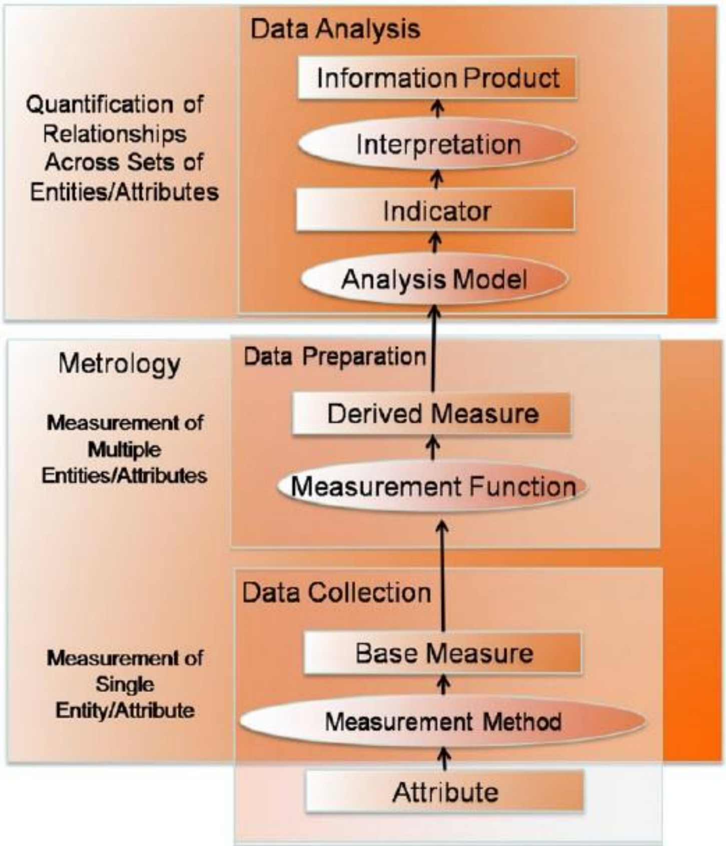


Figure 31 - ISO 15939 Measurement Information Model, obtained from Abran et al. [37]

In the data collection part of the process, which is the first metrology concept, a measurement method has to be defined for acquiring a *base measure*[2] (outcome of the measurement) of an entity's specific *attribute* [37]. As Abran et al. [37] emphasize, "[e]very base measure must correspond to a single, distinct, software attribute". As soon as the base measures are gathered, the data preparation phase is entered, where a *measurement function* is utilized to compute the *derived measures*. The measurement function is a mathematical or computational formula which receives as inputs at least two base measures and provides as an output a combined measure, i.e. the derived measure. The derived measures that need to be constructed are indirectly imposed by the *informational needs*. The accuracy of the derived measures relies on the accuracy of the base measures and the rigor of the mathematical formulas [37]. The data preparation phase is basically an intermediate phase before performing the quantitative analysis for supporting decision-making.

Data analysis begins with deciding how to model the relationships across entities and attributes in order to represent them with suitable *indicators* [37]. For this purpose an analysis model is defined. The final part of the ISO 15939 Measurement Information Model deals with the interpretation of the values provided by the indicators. The outcome of the interpretation step is the production of the *information product*. Information products are assigned or mapped to information needs as they relate to the satisfaction of the needs, while for the interpretation, the context of decisions or measurements has to be taken into account.

[2] All the terms in *italics* are defined in Table 23 (not all of them appear in Figure 31)

In the paper by Abran et al. [37], a three-step process is applied in two examples for determining the measurement method and measurement function as well as the base measures and derived measures related to an attribute. The three steps are the following:

1. **Determination of the measurement objective.** The objective of the measurement to be performed is defined as well as the attribute of the entity that the focus is on. The attribute(s) of interest depend on the objective.
2. **Characterization of the concepts to be measured.** The characterization of a concept should initially be based on the findings from a literature review. A concept may be an entity or a characteristic of it. By finding the way it is defined, the concept can be refined and decomposed into sub-concepts which entail the base measures.
3. **Assignment of Numerical Rules.** This step involves the determination of the measurement function for obtaining the derived measures.
    a. **Empirical description.** A measurement function is defined in a textual form, where the relation between the base measures is established.
    b. **Mathematical expression.** The empirical description is transformed into a mathematical expression which comprises the measurement function.
    c. **Measurement Scale Type.** The scale type of the derived measure is defined in order to establish how it will be used in statistical analysis or other mathematical calculations.

An important remark regarding measurement, though, is that, even if base measures and derived measures are measured adequately, there is uncertainty on whether these measures represent the concepts and relationships of the analysis phase [37].

In the ISO/IEC 15939 standard, various definitions are offered which are commonly used by the software measurement industry. As García et al. [92] have proved through their research, the terminology varies across different standards. Their extensive analysis and comparison of a wide range of software measurement standards and the definitions they provide has lead the authors in the conclusion that ISO/IEC 15939 is one of the more complete standards with respect to the number of terms covered, but some ill-defined terms still exist. The most important terms that need to be clarified, and that are relevant for this thesis as well, are presented in Table 23[3]. For more definitions and term comparisons, the reader can refer to the paper by García et al. [92].

[3] In the 'Definition' column of Table 23 two definitions are provided; the one suggested by the Software Measurement Ontology (SMO) [92] for harmonizing the differences among definitions and the one provided by the ISO/IEC 15939 standard.

Term		Definition
Indicator	SMO	A measure that is derived from other measures using an analysis model as measurement approach
	ISO15939	An estimate or evaluation of specified attributes derived from a model with respect to defined information needs
Measure		*Base measure*
	SMO	A measure of an attribute that does not depend upon any other measure, and whose measurement approach is a measurement method.
	ISO15939	Measure defined in terms of an attribute and the method for quantifying it. (A base measure is independent of other measures.)
		Derived measure
	SMO	A measure that is derived from other base or derived measures, using a measurement function as measurement approach.
	ISO15939	Measure that is defined as a function of two or more values of base measures.
		Measure
	SMO	The defined measurement approach and the measurement scale. (A measurement approach is either a measurement method, a measurement function or an analysis model)
Attribute	SMO	A measurable physical or abstract property of an entity, that is shared by all the entities of an entity class
	ISO15939	Property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means
Measurement Method	SMO	Logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale. (A measurement method is the measurement approach that defines a base measure)
	ISO15939	Logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale
Measurement Function	SMO	An algorithm or calculation performed to combine two or more base or derived measures. (A measurement function is the measurement approach that defines a derived measure)
	ISO15939	An algorithm or calculation performed to combine two or more 'base measures'
Analysis Model	SMO	Algorithm or calculation combining one or more measures with associated decision criteria. (An analysis model is the measurement approach that defines an indicator)
	ISO15939	Algorithm or calculation combining one or more base and/or derived measures with associated decision criteria
Decision criteria	ISO15939	Thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result
Information need	ISO15939	Insight necessary to manage objectives, goals, risks, and problems

Table 23 - Measurement terms definitions (related to ISO/IEC 15939)

4.3. Architecture Tradeoff Analysis Method (ATAM)

The Architecture Tradeoff Analysis Method (ATAM) belongs in the family of scenario-based software architecture evaluation methods which are described and evaluated by Babar and Gorton [93]. As the authors state, ATAM also applies attribute model-based analysis and is a mature method validated in various domains. Since ATAM deals with software architectures in general, it can also be useful in the analysis of the Enterprise Architecture as well.

ATAM is a concrete and formal method for evaluating an architecture's fit with multiple quality attributes through which implicit requirements could be discovered as well [94]. Thus, ATAM supports the clarification and refinement of requirements early in the process, the identification of conflicts among requirements [94] as well as the early identification of technical architectural risks [95]. Conflicts resolution can then be addressed though changes in the architecture [94], while the identified risks can be the focus of mitigation strategies [62] and be reused for further or future analysis [93]. Additionally, ATAM is a framework that promotes ongoing system design and analysis [94] and provides comprehensive support during the process [93]. For the analysis, ATAM does not indicate any specific evaluation techniques, but the trade-off analysis can include both quantitative analysis models as well as qualitative reasoning heuristics [93].

In the comparison of scenario-based software architecture evaluation methods carried out by Babar and Gorton [93], it is emphasized that ATAM is the only approach that takes into account multiple quality attributes, such as modifiability, security, performance, availability [94]. ATAM also facilitates the communication among stakeholders [94] as well as the understanding of the consequences of architectural decisions the stakeholders would make, since the decisions are linked and assessed with respect to not only the quality attribute requirements and measures, but also the business goals [62] [95]. Hence, ATAM helps in making trade-offs between competing attributes [93], but it does not guide economic trade-offs [62].

In Figure 32, the main outputs of ATAM are presented among which are sensitivity points and trade-off points. *Sensitivity points* are the components of the architecture which affect a quality attribute measure when a change in the decisions occurs [95]. The difference between the sensitivity points and the trade-off points depends on the number of attributes influenced by the decision [93]. Hence, *trade-off points* are the elements in the architecture where a decision change affects multiple attributes, either positively or negatively [94] [95]. The changes that usually take place can be local changes in the architecture, an internal change of a component not affecting other parts of the architecture or changes aligned to the architectural design and approach for expansion, for example [62].

ATAM includes four main areas of activities: (1) scenario and requirements gathering, (2) architectural views and scenario realization, (3) model building and analysis, and (4) trade-offs [94]. The last phase is where the critique takes place and the trade-off points are identified. According to Kazman et al. [94], this activity may result in:

- attribute models refinement and reevaluation,
- architecture refinement, model changes and reevaluation, or
- requirements changes.

ATAM consists of six steps which are presented in Table 24, based on the description provided by Kazman et al. [94], while in Figure 32 the inputs and outputs of the method are illustrated. More details about the output of ATAM can also be found in the paper by Kazman et al. [95].

Phase	Step	Description
Phase I	**1. Collect Scenarios**	• Requirements and scenarios are identified before the analysis process (any order is possible) • Operationalization of functional and quality requirements for gaining insight on the activities that the system should support
	2. Collect Requirements, Constraints, Environment	• Identification of attribute-based requirements, constraints and environment of the system (i.e. context identification) • Review of the previous scenarios to ensure alignment with the quality attributes
Phase II	**3. Describe Architectural Views**	• Generation (design) of candidate architectures • Description of them in terms of architectural elements and properties related to each quality attribute • Building and maintenance of attribute models (qualitative and quantitative) for facilitating the *reasoning* about the architecture and any changes that lead to new architectures. These models are the utility trees of the quality attribute requirements
Phase III	**4. Attribute-Specific Analyses**	• Separate performance analysis of each quality attribute with respect to each architecture based on performance scenarios (evaluation of attributes in *isolation*) • Calculation of the degree that a scenario meets the requirements defined in Step 2 • Results: description of system behavior regarding values of the attributes Thus, strengths and weaknesses of the system and its components can be determined.
Phase IV	**5. Identify Sensitivities**	• Determination of sensitivity of quality attributes with respect to architectural elements: change in the attributes, leading to change in the models and consequently in different results. Focus on primary attributes is possible; thus, parallel analysis can be avoided and analyses can be performed based on priority.
	6. Identify Tradeoffs	• Evaluation of the models built in Step 4 for identifying tradeoff points • Use of sensitivity points for determining the tradeoff points; tradeoff points are the elements that constitute a sensitivity point for multiple attributes

Table 24 - ATAM steps

As ATAM is often combined with CBAM in literature and in practice, Nord et al. [62] have proposed some refinements of the ATAM steps in order to improve the interaction of the two approaches. For example in Step 2 of ATAM, the authors propose the elicitation of additional business-related information, such as dependencies and priorities of business goals as well as their schedule and cost constraints as imposed by business drivers. Furthermore, they indicate that the identification of relations between business goals and quality goals would be beneficial for applying the CBAM.

Thus, ATAM provides an analysis of different scenarios of the architecture with respect to the quality attribute requirements and makes an assessment of the architecture based on how sensitive are the elements of the architecture when a decision is changed. This specific characteristic in combination with the relation of the quality attributes with business goals, as proposed by Nord et al. [62], make ATAM useful and applicable for the analysis of Enterprise Architectures as well.

Figure 32 - ATAM inputs and outputs

4.4. Goal Contribution Change Analysis

Changes in a system are derived from changes in the environment of the organization, but the changes are reflected on a system or a process only when the change in the business environment causes a change in the stakeholders' goals and requirements for adjusting to the environment [96]. A goal or a requirement change may entail the addition, modification or deletion of a goal or requirement respectively [97]. A change in the goal graph can have an impact on the dependent elements of the modified goal and on the dependency relationships themselves [97]. A way to denote the dependency relationships is through contribution links which represent the degree of one goal influencing another, either positively or negatively. In the goal contribution analysis, two types of changes can occur according to Teka [98]: (i) change in goal satisfaction levels and (ii) change in contribution relationships.

Ellis-Braithwaite et al. [99] have pointed out the three approaches for assigning scores on contribution links in goal graphs:

- Subjective qualitative scores, e.g. --, -, +, ++
- Subjective quantified scores, e.g. -100 to 100
- Objective measure variable, i.e. a measured quantity predicted to be increased, reduced, etc.

Among the three approaches, the objective measure variables are preferred since they add "rigor and testability to the task of deciding between alternatives, [while] the same applies to the task of demonstrating alignment to business objectives" [99].

Furthermore, Ellis-Braithwaite et al. [99] state the importance of expressing the satisfaction of a requirement in a quantitative manner comparing to a qualitative one. Qualitative descriptions are ambiguous in terms of how to achieve the requirement and of how to analyze its impact on the strategic level. An exception is, though, made for qualitative terms defined as fuzzy numbers.

In the area of analyzing contribution relations among goals and requirements, a few methods have been identified in literature. These techniques are presented in APPENDIX IV. The focus of the described methods is upon managing the dynamic nature of stakeholders' goals in goal graphs, by describing techniques that capture changes through contribution or dependency links. Such techniques facilitate the decision-making process, the allocation of resources to goals and the reflection of the impact of change in the goal level to the EA level.

These techniques are: (i) the *NFR based Fuzzy Logic Approach* developed by Teka in his thesis [98] and also presented in the paper by Teka et al. [96] which focuses on change impact analysis on indirect contribution relationships of goals and requirements, and (ii) the *DepRVSim* developed by Wang et al. [97], which stands for Requirement Volatility Simulation considering Dependency relationships and is a simulation approach for modeling the dependency and traceability relationships among requirements and the changes that occur in the structural dependency relationships. Additionally, AGORA is a method developed by Kaiya et al. [100] for supporting the existing goal-oriented analysis techniques by adding attributes, such as contribution and stakeholder's preference values, on the goal graphs and, hence, improving the goal and requirements refinement and decomposition tasks. The last method does not deal with change primarily, but it aims to identify the most suitable decomposition of goals and requirements by using contribution links.

4.5. Goals and Performance Indicators

A strong relation between goals and performance indicators has been identified in literature in an attempt to quantify goals and their satisfaction levels. In this section, characteristics of goals that can support measurability are presented, performance indicators are introduced and their relation with goals is clarified. Furthermore, methodologies for relating goal models or structures and indicator representations are discussed.

4.5.1. Goals properties: facilitating measurability

Defining a goal in a formal way, as it has been pointed out by many researchers, is important, especially when the goals are going to be measured. Goal analysis approaches, such as GQM, GQ(I)M, GQM$^+$ Strategies, have introduced and considered templates for formal definition of goals, either high-level, business goals or low-level, measurable goals. Schneider et al. [101] have conducted a literature research regarding goals in the Enterprise Architecture Management context, where they identified all the properties of a goal that should be included in a proper definition. The properties are presented in Table 25.

1. Name of the goal ***2. Category***, based on criteria such as: • Time horizon • Degree of Measurability • Area of origin (e.g. customer-related, IT-related) ***3. Rationale***, denoting the reason for establishing a goal ***4. Stakeholder*** ***5. Direction***, reflecting the goal's intention: e.g. increase, reduce, facilitate, etc. ***6. Focus area***, indicating the EA layer or domain that the goal refers to or the exact element that it influences, e.g. process, service, resource, etc.	***7. Restrictions*** comprise an extension to the focus area, since they provide more information for the applicability of the goal. ***8. Target value*** is a property that complies with the SMART[4] concept and comprises the desired qualitative or quantitative value or the absolute or relevant change of the current value. ***9. Deadline*** (in line with SMART) is the point in time when the goal is expected to be achieved. ***10. Implications*** encompass the consequences of the success or failure of the goal. ***11. Interrelations*** cover the relationships between goals, such as generalization, specification, conflict, etc.

Table 25 - Goal properties

Defining goals in terms of these properties can enhance their concreteness and measurability. Attributes regarding the 'target value' and the 'deadline' impose clear instructions for achieving the goal, while the 'focus area' or 'interrelations' attributes facilitate traceability and the linkage with the indicators or measures, as it is explained in the rest of this section.

4.5.2. Key Performance Indicators (KPIs)

KPI Properties and Measures

Key Performance Indicators (KPIs) are a subset of performance indicators defined by an organization for monitoring its progress. The indicators included in the KPIs subset can provide a representative picture of the organization's performance and bear a reasonable burden of costs for the organization regarding their measurement and monitoring [102]. KPIs comprise financial and non-financial metrics which aim to "quantify the performance over time toward the meeting of [the organization's] strategic and operational goals" [103].

Cardoso [103] has recently made a research on how KPIs are related to the measurement of goals in EA with respect to the support for representing KPI-related concepts in formal modeling languages and for measuring goals' achievement. Cardoso has indicated that goal modeling approaches and languages do not provide enough support for modeling KPI-related concepts and associating them with goals. Thus, representation of KPIs is considered separately.

KPIs can be defined in terms of a number of attributes, some of which are aligned with the goal attributes mentioned in the previous section (Section 4.5.1.). As for the goals, KPI's attributes are usually associated with the SMART criteria for their definition [103]. The most common attributes of a performance indicator that meet the SMART criteria are the following: ***name, description, scale, measure, current value, target value, threshold, source, frequency, responsible, informed,*** and ***owner*** [103].

[4] SMART is a mnemonic for defining criteria and stands for *Specific*, *Measurable*, *Achievable*, *Relevant* and *Time-bounded*. It is also used for defining objectives and KPIs [103].

'Source' is the attribute that contains the internal or external sources from which the performance indicator is derived, such as company policies, mission statements, business plan, job descriptions, laws, domain knowledge or the systems where the data comes from for the calculation of the indicator [102] [103].

Popova and Sharpanskykh [102] have also included ***'hardness'*** as a parameter when defining an indicator, which can take two values, soft and hard. *Soft* is an indicator which is qualitative and not directly measurable, such as customer satisfaction, while *hard* is a quantitative indicator which can be directly measured or calculated. Measuring a soft indicator can be achieved by relating it with hard indicators. Popova and Sharpanskykh suggest, for instance, the use of the percentage of returning customers, the percentage of on-time deliveries or the number of complaints, as indicators for measuring customer satisfaction. The selection of the most suitable hard indicators for demonstrating the state of a soft indicator is company-specific [102].

Cardoso [103] also indicates which of the KPI attributes are in line with the SMART criteria (Table 26).

SMART	Attribute	Description
Specific and **M**easurable	**Measure**	Defines how the KPI values are calculated by measuring the attributes of the EA elements. The measurement results in the ***'current value'*** of the KPI. The ***'scale'*** is also directly related to the specification of the performance indicator, since it determines the unit of the expected measurement.
Achievable	**Target Value**	Represents the ideal desired value. It must be compared with the *'current value'* in order to designate the state of the indicator and the deviation that exists. ***'Threshold'*** is the attribute used for defining the acceptable deviation. Various types of target values exist. For further details, the reader can refer to Cardoso's research [103].
Relevant	**Goal**	The goals that the KPI is related to.
Time-bounded	**Scope**	***'Frequency'*** is the attribute that denotes the frequency of the measurement during a specific *time interval*. The ***'time frame'*** is a property of a performance indicator which denotes the length of the time interval for which the indicator will be evaluated [102]. The scope also contains the linkage of the KPI with the organizational structure through the last three attributes (***responsible***, ***informed***, and ***owner***).

Table 26 - SMART criteria and KPI attributes

Even more contributing to this thesis is the detailed analysis conducted by Cardoso [103] with respect to the measurement of KPIs and the most important dimensions considered regarding the 'measure' attribute when performance analysis is related to EA elements (Figure 33).

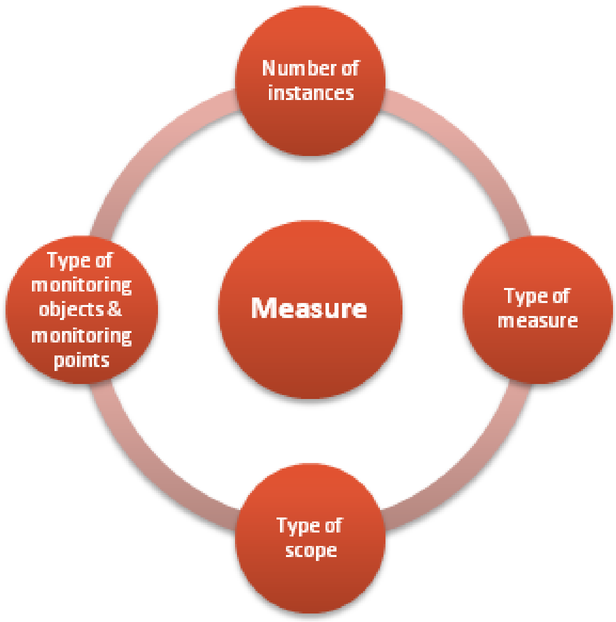


Figure 33 - Measure dimensions

The ***number of instances*** dimension in Figure 33 refers to the number of instances that are necessary for calculating the KPI. From this perspective, there are three types of measures: base measures, aggregated measures and derived measures [103]. The base measure and derived measure types were also discussed during the description of the ISO 15939 Measurement Information Model. The ***type of measure*** dimension entails the type of the property to be measured, e.g. time, count (number of times something occurs), condition, data-related, human resource-related [103] or continuous/discrete measure [102]. The ***type of scope*** dimension filters the process instances and selects those to be used for the calculation of the KPI values. Finally, the ***type of monitoring objects*** dimension refers to the definition of the EA elements whose properties are examined, such as processes, data objects, human resources, software and hardware. The ***monitoring points***, consequently, function as the points where the data is gathered and becoming available for computations [103]. Thus, monitoring points can be attached to any object of the architecture that has the responsibility of data gathering.

For representing and relating KPI-related concepts, the relations among KPIs should also be defined. This observation leads to the following section (Section 4.5.3.) where research on performance indicator (PI) structure models and systems is presented. Identifying the relationships among KPIs influences their measurements as well. Cardoso [103] has identified a set of relationships that support this stream of research. The types of relationships mentioned in her paper are: *correlation*, *causality*, *conflict*, *independence*, *trade-off*, *costlier_than* and *customized*.

Except from Cardoso, recently another group of researchers have been dealing with the definition of EA-oriented KPIs and the elements regarding their proper determination. Matthes et al. [104] have introduced a structure for defining, documenting and retrieving EA Management (EAM) KPIs based on the EAM goal list and the EAM KPI catalog proposed by the same author group in 2011 [105]. Matthes et al. [106] have also designed a method for defining EAM KPIs based on the EAM goal list and the EAM KPI catalog. Their motivation for developing the EA Management KPI Structure is that it would ensure KPIs comparability, foster reusability and guide the development process [106].

The EAM KPI Structure consists of two types of KPI structure elements, which were also evaluated by the authors through an online expert survey and mapped to KPI descriptions found in literature [104]. An example of the EAM KPI Structure is provided in APPENDIX III (Figure 114). The two types of structure elements are:

1. **General structure elements:** ***title***, ***description***, related ***goal(s)*** from the EAM goal list, ***calculation rule*** or way to compute the KPI, ***source*** of the calculation rule of the KPI (literature or practice), ***layers and cross-cutting aspects*** (parts whose input is required for the calculation, based on the 'Cross-cutting aspects of EA' model proposed by Buckl et al. [40]), ***information model*** (entities, relationships, entities' properties required as input for the computation in the form of a UML diagram, for instance), ***code*** (the ID of the KPI).
2. **Organization-specific structure elements:** This part of the structure consists of two tables:
    a. **Mapping table:** Indicates the link between the elements of the information model and the organization-specific concepts. There are four columns in this table: ***name in model*** which is a list of the names that appear on the information model, ***mapped name*** provides the link with the corresponding organization-specific concepts, ***contacts*** indicates the data owner and ***data sources*** comprises the technical organization-specific data storage.
    b. **Properties table:** In this table, eight additional properties of the KPI are described and documented. It consists of three columns; the first includes the properties' ***name***, the second the ***property value*** which applies for the organization and the third the ***best-practice***, which indicates the common values observed in practice. The properties included in this table are: ***measurement frequency***, ***interpretation*** of possible KPI values, ***KPI consumer***, ***KPI owner*** (responsible for the achievement), ***target value***, ***planned value(s)***, ***tolerance value(s)***, ***escalation rule*** (guidance for handling uncontrollable events).

From the remarks resulting for the surveys, Matthes et al. [104] have pointed out that some of the experts considered the EAM goal list quite unspecific and general, and they suggested more goals to be added. The EAM goals can function as a general guideline for relating KPIs to areas of interest as well as high-level business goals which can be further refined and decomposed into more specific and measureable ones.

Consequently, Matthes et al.'s categorization comprises a classification of the KPI properties found in literature into general and organization-specific, while the notion of mapping the exact information model related to a KPI is also valuable for providing evidence regarding the origin of the data used for the computations.

Table 27 provides the mapping between the properties discussed by Matthes et al. [104] and by Cardoso [103]. The red columns and rows indicate the existence of a mismatch. The following observations occur:

- **'Current value'** mismatch: The current value is very essential for evaluating the progress of the performance of the organization, thus its absence from the KPI structure is considered as a deficiency.
- **'Responsible'** and **'Informed'** mismatch: These attributes are related to the **'KPI consumer'** element, so the interchange of the terms or the decision of the most relevant one when defining an indicator is acceptable.
- **'Timeframe'** mismatch: the timeframe is also important for indicating the measurement period and the evaluation period. Thus, it can be derived from the **'planned values'** which indicate the future expected values of the KPI.
- **'Layers'** mismatch: Since the focus of the KPIs in both research papers is on EA, the layers and cross-cutting aspects to which the KPIs refer should be considered. This property can also be combined with the 'type of monitoring objects and monitoring points' dimension of the measure (Figure 33). In addition, there is a strong relation between the latter and the 'information model' of the KPI general structure elements.
- **'Interpretation'** mismatch: This element is not exactly a KPI property, but it is rather useful when an evaluation of the performance of the KPI occurs. This observation is also verified by the methodologies discussed in the next section (Section 4.5.3.).
- **'Scale'** mismatch: The scale of the indicator is an attribute that should be determined precisely, as it can contribute to the calculations and transformations needed for resulting in the indicators value. It is not clearly defined in the EAM KPI Structure, but it is probably included in the **'calculation'** structure element.

	EA Management KPI Structure [104]														
	General structure elements							***Organization-specific structure elements***							
	Title	Description	Goals	Calculation	Source	Layers	Code	Measurement Frequency	Interpretation	KPI consumer	KPI owner	Target value	Planned value(s)	Tolerance value(s)	Escalation rule
KPI attributes [103]															
Name	x														
Description		x													
Scale															
Measure				x											
Current value															
Target value												x			
Threshold														x	
Frequency								x							
Source					x										
Responsible															
Informed															
Owner											x				
Goal			x												
Timeframe															

Table 27 - Mapping of KPI Attributes to EAM KPI Structure Elements

In addition, there is a stream of research concerning performance indicators which focuses on indicators specialized in business processes. These performance indicators are called Process Performance Indicators (PPIs), which are defined by del-Río-Ortega [107] in her PhD Dissertation as “quantifiable metrics that allow to evaluate the efficiency and effectiveness of business processes”. del-Río-Ortega [107] has also introduced the PPINOT metamodel for relating PPIs with business process elements through other relevant elements in order to ensure traceability. As it is also emphasized by del-Río-Ortega et al. [108], PPINOT is independent of the business process modeling language, since it is based on an abstract business process modeling language that can easily be mapped to any language used [107]. The attributes of PPIs are a subset of the KPI attributes discussed in this section, as it is illustrated in the metamodel in Figure 34. Moreover, del-Río-Ortega [107] discusses in her PhD dissertation, the nature of measures which is in line with the ‘type of measure’ dimension described by Cardoso [103]. The ones identified by del-Río-Ortega are: time, count, condition and data.

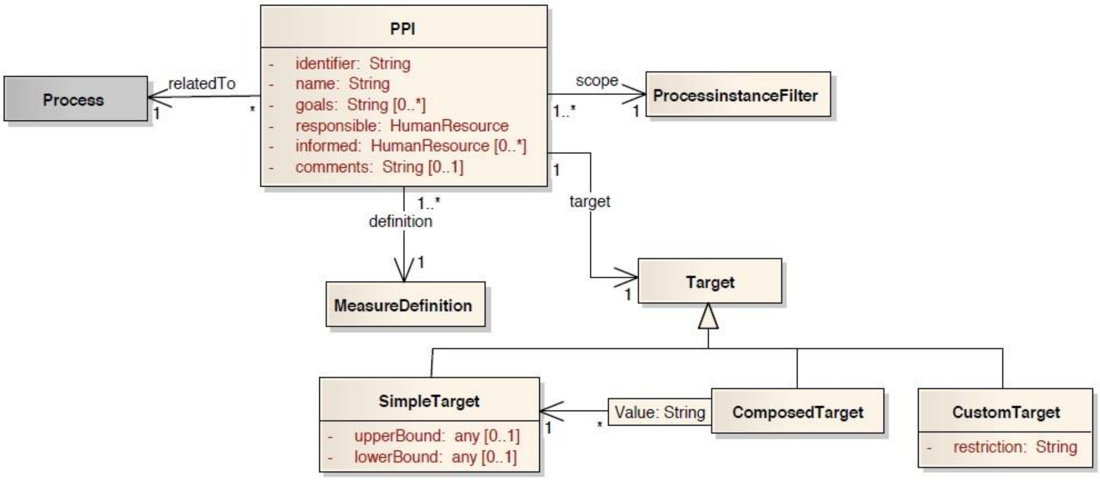


Figure 34 - PPIs in PPINOT metamodel, obtained from del-Río-Ortega [107]

Visualization of KPIs

While descriptions regarding KPI properties and templates for documenting them are available in literature, attempts for visualizing the indicators are more limited. As del-Río-Ortega [107] states in her dissertation, there are models for visualization of business processes, and similarly for visualization of goals and enterprise architectures, but performance indicators definitions are restricted to textual documentations and lower level languages. Thus, the visualization of KPIs can bridge the visual gap between the EA or goal models and the KPI definitions (this statement is a generalization of the statement made by del-Río-Ortega [107] regarding business process models and PPI definitions). Additionally, visualizations can also improve decision-makers’ understandability of complex issues.

Following the last two statements, del-Río-Ortega [107] has introduced a graphical notation of PPINOT which includes representations for three measure dimensions: number of instances, type of measure and type of scope. Additionally, she has proposed ‘connectors’, which are ways for linking measures. Through this notation, del-Río-Ortega manages to represent a PPI, or in general a performance indicator, in terms of the measures that are needed for deriving its value, while capturing the formulas used for the calculations. Next, the PPIs are directly assigned to the processes in the business process model. No links among indicators are provided, though. An example of the PPINOT graphical notation and the representation it provides is illustrated in Figure 35.

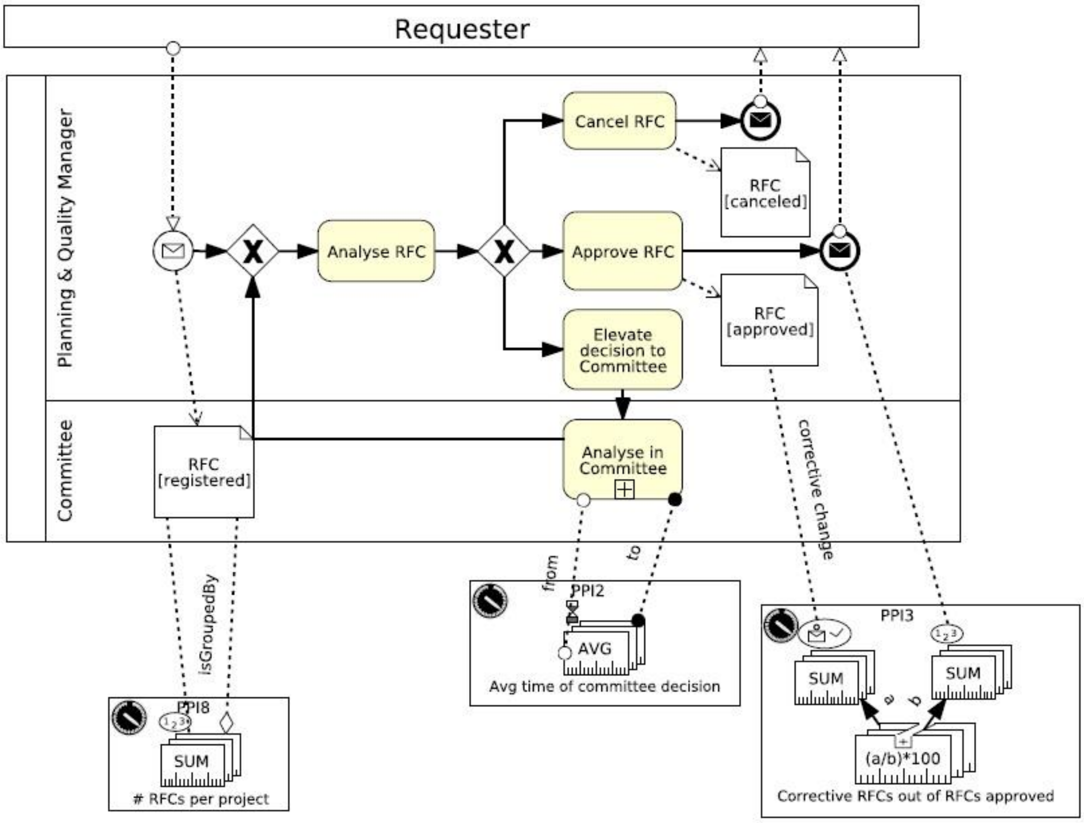


Figure 35 - Graphical representation of PPIs based on the PPINOT Graphical Notation, obtained from del-Río-Ortega [107]

A different visualization method has been proposed recently by Rojas and Jaramillo [109]. Rojas and Jaramillo proposed the use of executable pre-conceptual schemas for representing KPIs and their respective attributes. The schema comprises a graphical representation of the information surrounding a KPI in order to foster the visualization of the KPI and enhance the processing and analysis of the information. More concepts are included in the schema, such as processes, roles, intentions that support the calculation and the definition of the indicator. Figure 36 shows the pre-conceptual schema, where the properties of the KPI discussed in the previous paragraph and even more detailed information regarding the KPI are depicted in the right side of the schema. Figure 37 illustrates an example of the schema after its execution, where a selection of the most relevant attributes has taken place. Such a schema could be a useful replacement of extensive documentations regarding the definition and data extraction of KPIs. It is rather difficult, though, to visualize the relation with other indicators on the schema, apart from the reference to them at the formula property.

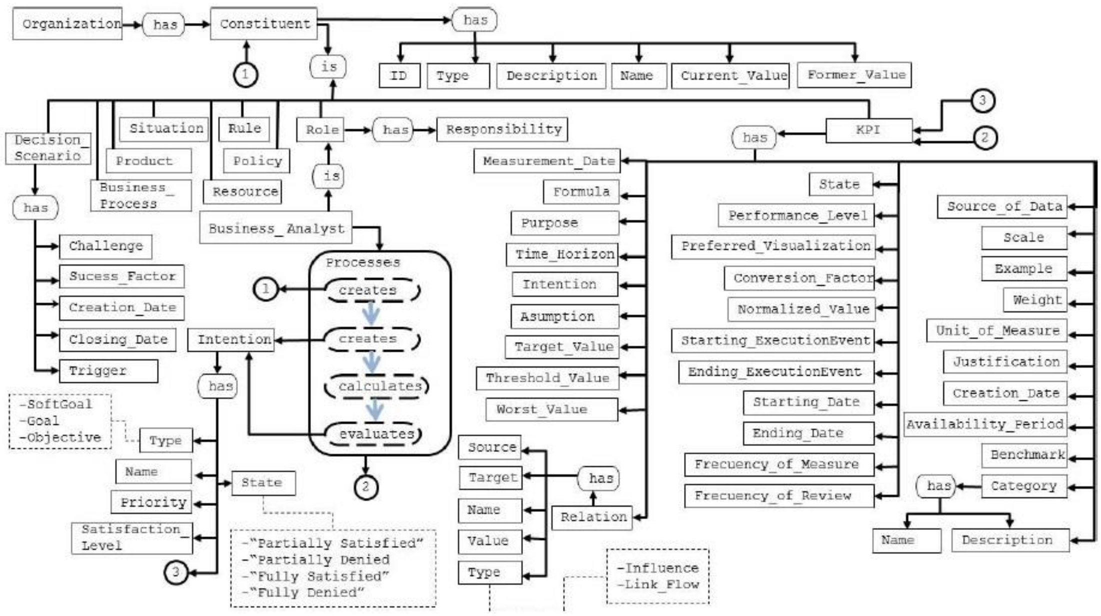


Figure 36 - Pre-conceptual schema for representing KPIs, obtained from Rojas and Jaramillo [109]

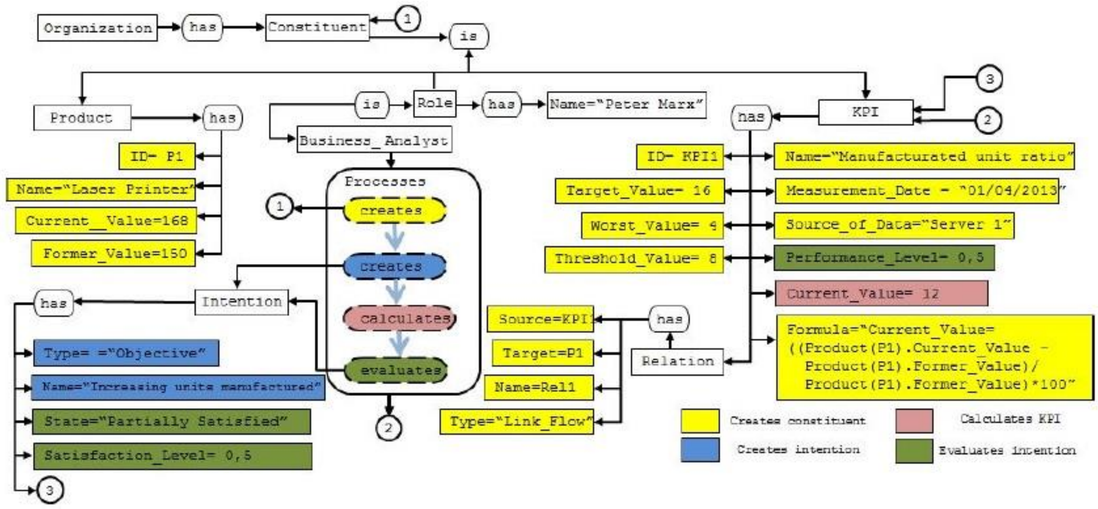


Figure 37 - Executable example of the Pre-conceptual schema, obtained from Rojas and Jaramillo [109]

4.5.3. Performance Indicators Structure Models and Value Propagation

Structured graphs have been considered, in general, as a method to avoid ambiguity and misunderstanding, and to gain insight on the dependencies among the modeled concepts by representing and visualizing their interrelations. In the previous chapter, structures have been discussed regarding, for instance, software development components and resource allocation, while in this chapter the focus is on goal graphs and models. Accordingly, research has been conducted regarding performance indicator structures, how they are related to goals and how the values of the indicators are propagated in such graphs.

4.5.3.1. Goal Structures and Performance Indicator Structures

Popova and Sharpanskykh [102] have developed a framework which illustrates the relationships among the core aspects of structure and behavior of the organization and groups them in four main views (process, performance, organization, and agent). The focus of their research is the performance-oriented view which incorporates a *goal structure*, a *performance indicators (PI) structure* and relationships between them which are achieved through the use of *goal patterns* and *performance indicator expressions*. For each structure type, a set of relationship types are described which allow the representation of complex relationships among objectives and performance measures. A simplified graph of the performance-oriented view elements is presented in Figure 38, which is made based on Figure 2 and Figure 1 in Popova's and Sharpanskykh's research papers [102] and [110] respectively.

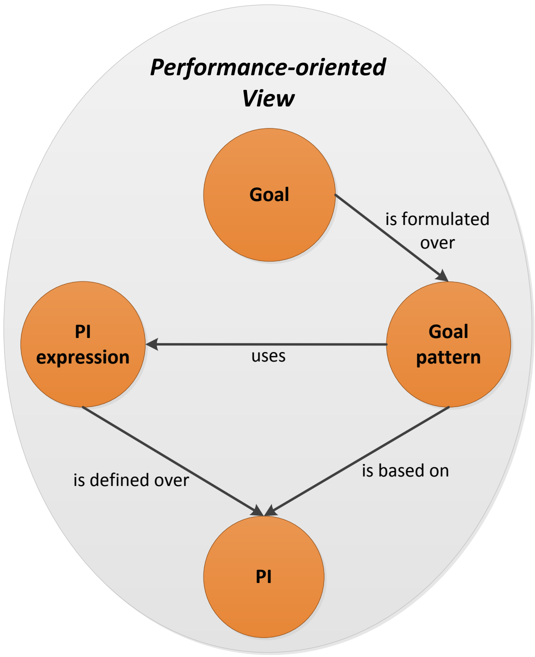


Figure 38 - Performance Indicator structure & Goal structure in Performance-oriented view

Goal patterns are defined as "properties that can be checked to be true or false for the organization, unit or individual at a certain time point or period" [102]. For determining goal patterns, performance indicator expressions are used, as it is illustrated in the figure. A *performance indicator expression* is "a performance indicator or a mathematical statement over a performance indicator containing >, ≥, =, < or ≤. A performance indicator expression can be evaluated to a numerical, qualitative or Boolean value for a time point, for the organization, unit or agent" [102]. There are three types of goal patterns, which are matched with certain types of PI expressions, as Popova and Sharpanskykh [110] proposed:

Goal Pattern type	PI Expression type
1. achieved/ceased **2. maintained/avoided**	*Boolean,* where the values are evaluated as true or false
3. maximized/minimized	*Numerical* or *qualitative*, where the values can be ordered

Thus, the goal patterns and PI expressions provide the logical connection between the goals and performance indicators. More precisely, goal patterns provide meaning to the various value domains that result from the divisions that PI expressions imply, and enable the reflection of organizational performance that should be achieved or maintained [102]. Furthermore, goals are considered, by Popova and Sharpanskykh [102], as objectives which have as a basis the goal pattern extended by desirability and priority information. This strong interrelation between goals and performance indicators indicates that there should be a similar interrelation between their structures as well.

For building a *structure of performance indicators*, relationships between the indicators have to be defined and the sources for deriving the relationships have to be determined. Popova and Sharpanskykh [102] recommend a set of possible sources, while as far as the relations are concerned they have focused on and formally described three relation types among indicators:

- **Causal relation:** one performance indicator causes change in the same or opposite direction to another indicator. The level of influence is also specified in the relation.
- **Correlation:** change in one indicator results in positive or negative change to another indicator (positive or negative correlation).
- **Aggregation relation:** when two indicators are related through an aggregation relation, it means that they are the same measure at a different aggregation level. The aggregation relation also entails positive correlation between the indicators and the same type and unit.

A simplification, suggested by the authors, comprises the replacement of all relation types with causality relations [102]. Inference rules based on causality relationships can be also used for deriving unspecified relations between indicators. Inference rules have been determined for the level of influence that corresponds to the new causal relation (see Table 1 in [102]). In addition, a performance indicator's causal relation with another performance indicator can be derived from the refinement relation between their corresponding goals by following the rules below (based on the description by Popova and Sharpanskykh [102]):

Goals: Refinement relation	$\longrightarrow$	**PIs:** Causality relation
Goals: Refinement relation *AND* **[PI expressions:** contain comparisons (e.g. >, <) or qualitative measures *OR* **Goal Patterns:** increase/decrease **]**	$\longrightarrow$	**PIs:** specific type of causality relation (e.g. positive, negative)

A representation of the resulting PI structure can be found in Figure 39.

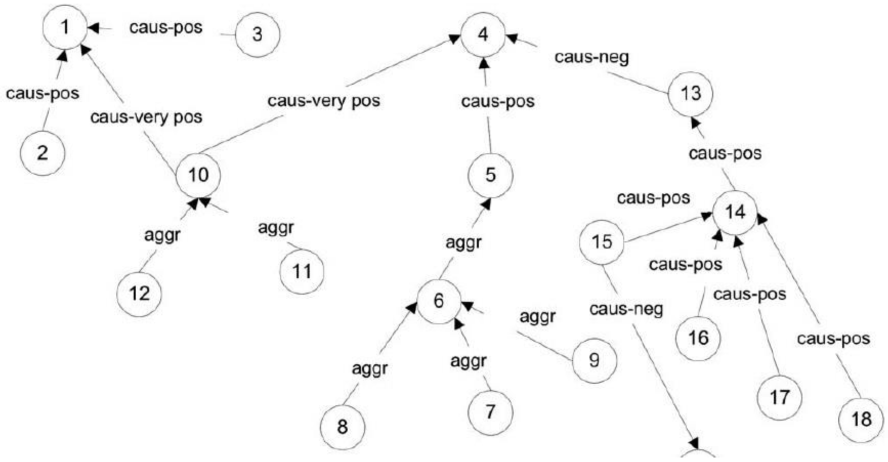


Figure 39 - PI Structure example, obtained from Popova and Sharpanskykh [102]

Popova and Sharpanskykh [102], in the beginning of their paper (p. 506), suggest that a formal definition of goals starts with the determination of the performance indicators and PI expressions, continues with the definition of goal patterns and ends with the goal specification; while, as their analysis continues, the authors indicate that the creation of the *performance indicators structure* and the *goal structure* should be simultaneous due to mutual influences and interrelations (p. 516). Two alternatives are discussed for accomplishing the simultaneous development of the two structures, which are similar to the two directions described by Frank et al. [111] for designing indicator systems. Additionally, a relation with other methodologies has been identified and presented in Table 28.

Frank et al. [111]: *Identification of indicators*	**Popova and Sharpanskykh** [102]: *Parallel building of PI and goal structures*	**Similarities with other methods**
Top-down direction: operationalization of business goals and strategies, by asking the question 'What do we want to measure?'	**Starting with objectives** extracted from company documents, formulating them into goals during the design process and crystallizing the corresponding performance indicators.	Both approaches are aligned with **GQM-related methods**, since there is a need for refining the goals until there is a direct reference to their measurement.
Bottom-up direction: identification of measurable concepts, by asking the question 'What can we measure?'. This direction indicates starting from available data and already known indicators.	**Starting with key performance indicators** and their related measures, deriving the desirable values of the indicators leading to general goals, followingly to formal PI expressions and goal patterns, and finally to goals by adding other characteristics, such as priority and desirability.	

Table 28 - Identification of Indicators, Development of PI structures & Similarities with other approaches

By considering the two alternatives as well as the initial suggestion by the authors for starting the process with performance indicators and PI expressions for defining the goals, it can be inferred that there is a high complexity in defining goals and the related performance indicators. Therefore, special attention should be placed upon the structures for clarifying the measurability of goals as well as their satisfaction. The organizational performance can then be inferred based on the achievement of goals.

In terms of the model-to-model transformations from the Design space to the Analysis space, which were explained in Section 3.2., the performance indicators structure could function as the Goal Analysis Model in the Analysis space. Since the Goal Analysis Model is derived from the Goal Design Model, in the terminology of structures, the Goal Design Model is the goal structure, while in the terminology of ArchiMate® it is the motivation model. Accordingly, Figure 10 can be refined into Figure 40. By using the PI structure in the Analysis space, the link between the Architecture Analysis Model and the PI Structure becomes more explicit, since there is a direct relation between the measures necessary for computing the indicators and the elements of the architecture from which the input data is taken.

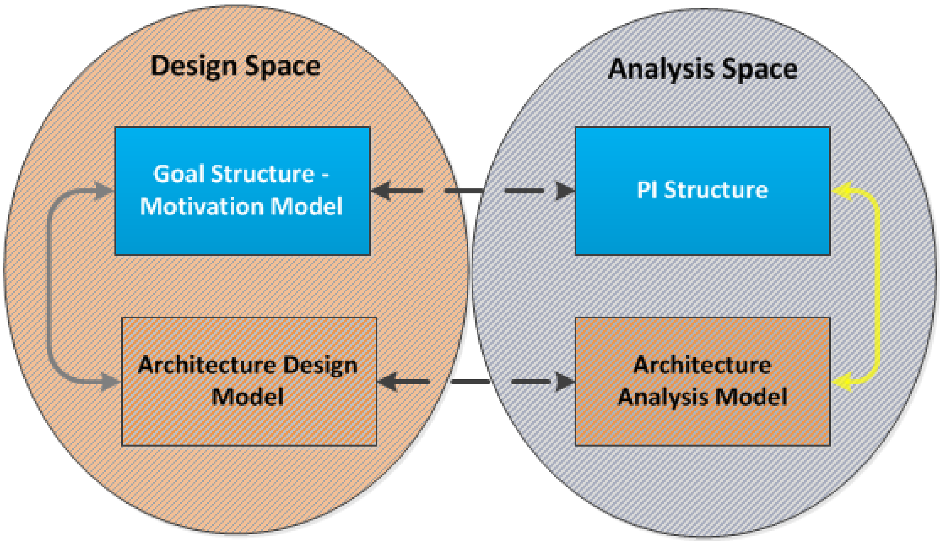


Figure 40 - Refinement of Analysis Space: PI Structure

Frank et al. [111] have also introduced a Performance Modeling Language, called SCORE-ML, as part of a process framework which covers the entire lifecycle of indicator systems ('design', 'use' and 'refinement' phases). SCORE-ML associates indicators with goals and provides relationships between indicators as well. The relations between indicators, considered in the paper by Frank et al., are mostly based on computation dependencies, while another vague type of relation is also defined, the 'similar to' relation. By relating indicators based on their calculation formulas, another kind of structure can be created. By following the graph with the calculations performed for setting a value to an indicator, there are three feasible benefits:

1. tracking of the data needed for the calculations
2. identifying the elements in the architecture which are more valuable for the calculations because of the importance of the data that come from them, and
3. monitoring effects on indicators when elements of the architecture are changed.

In such an indicator structure, the lower-level indicators can be viewed as the *base measures*, while the higher-level ones as the *derived measures* in terms of the 'number of instances' dimension of the KPI's 'measure' attribute. Figure 41 demonstrates the computation relation through the circled asterisk.

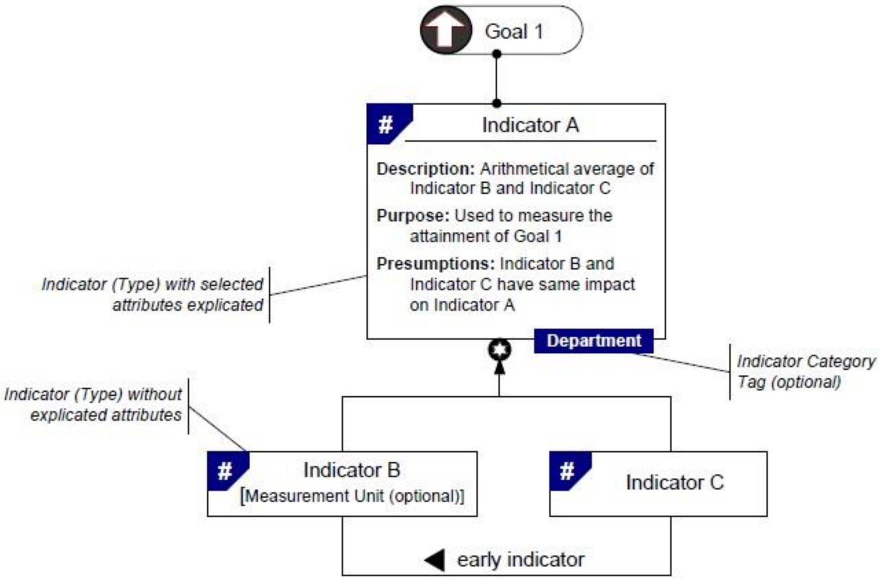


Figure 41 - Graphical notation of Indicator Computation Relation, obtained from Frank et al. [111]

4.5.3.2. KPIs Computation and Value Propagation

As KPIs are associated both with goals and elements, such as processes, models have been proposed which aim to combining the two views (goals-indicators and elements-indicators). This combination attempts to provide support for business intelligence activities. Performance indicators' satisfaction values and performance levels can be translated into or reflected upon goals' satisfaction values and enhance performance analysis of the organization.

Barone et al. [112] and Mate et al. [113] propose the Business Intelligence Model (BIM) for modeling the business strategy. In their process model they include business goals, indicators for assessing the goals and situations (or events in the terminology of ArchiMate®) which can influence goals. A situation is also related to an indicator. As Mate et al. [113] emphasize, BIM represents the SWOT analysis (Strengths, Weaknesses, Opportunities and Threats) and "provides a comprehensive view of the business strategy along with KPIs and their relationships".

KPIs in BIM have a double role expressed by the two possible relationships; the 'evaluates' relation which indicates that the KPI assesses a specific goal and the 'measures' relation which connects the KPI with the business process considered responsible for the realization of the lowest level goals [112] [113]. While BIM was developed in the context of reasoning regarding business processes, in the context of ArchiMate®, the element related to an indicator through the 'measures' relation can be of any kind depending on the indicator and the data that needs to be obtained. Moreover, a subset of the indicator characteristics are considered by Barone et al. [112] for calculating the *performance level* of a KPI. The performance level (pl) is a normalized value, ranging from -1 to 1, which results from the evaluation of the *current value* of the indicator against the *target value*, the *threshold* and the *worst value*. Performance regions are also derived from the indicator characteristics, while mapping the current value of an indicator to a performance level and consequently a performance region is a result of a linear interpolation based on the following equation:

KPI Performance Level Equation

$$pl = \begin{cases} \dfrac{|current\ v. - threshold\ v.|}{|target\ v. - threshold\ v.|}, & if\ current\ v. \geq threshold\ v. \\ \\ -\dfrac{|current\ v. - threshold\ v.|}{|thershold\ v. - worst\ v.|}, & if\ current\ v. \leq threshold\ v. \end{cases}$$

The relation of the indicators in BIM and the mapping of performance levels of indicators are illustrated in Figure 42.

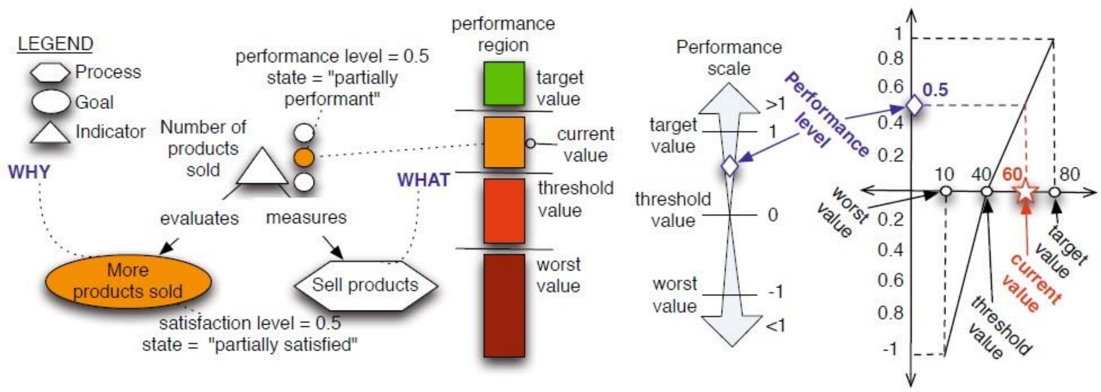


Figure 42 - BIM Indicator Performance regions and Performance levels, obtained from Barone et al. [112]

In addition, BIM includes algorithms and techniques for assigning values to composite indicators through propagation in a bottom-up fashion. Barone et al. [112] consider as trivial the situation where a mathematical function exists for representing the relation among indicators; hence, they focus on the cases when "indicator values have to be derived using estimation or approximation techniques". An Eclipse-based prototype tool has also been developed for supporting the BIM techniques.

There are three techniques described by Barone et al. [112] for deriving composite indicators' values, from which the first two are quantitative and the third is qualitative. Availability of data and time of the expert who designs and monitors the performance determine the selection of the technique, while the qualitative technique can indicate conflicts and inconsistencies in the early design phase. The two quantitative techniques are the following:

1. **Use of Conversion Factors.** Conversion factors are used when the indicators that need to be combined do not share the same unit of measure. Such a factor is basically a multiplier for eliminating the differences among indicators and enabling mathematical calculations among them. The formulas can also include weights besides the conversion factors. Thus, an indicator could be expressed as:
$$i_{g1} = i_{g1}^{e} + w_{g2} \cdot c_{g2} \cdot i_{g2} + \cdots$$
where i_{g1} is the composite indicator, i^{e}_{g1} is the expected value of the indicator, g is a goal, w is the weight of the component indicator, c is the conversion factor and i_{g2} the component indicator assigned to a sub-goal. A similar pattern is used when an

indicator from an influencing situation (event) is one of the component indicators. Thus, there are two types of indicator sources: from influencers and from sub-goals.

2. **Use of Range Normalization.** This method is used when conversion factors cannot be defined. The normalization technique receives as input a set of values spanning a specific range and transforms them into another range (-1 to 1) for representing the performance level of the indicators. Different performance levels of the component indicators can be combined into the level of the composite indicator. Weights are used for the calculations as in the conversion factors technique. Possible functions for calculating the composite performance level are: sum, min, max, average. An example of propagating the performance level of the component indicators to the composite indicator is the following:

$$pl_{g1} = sum\left\{ w_s \cdot pl_{i_s}, \max\left[pl_{g2}, pl_{g3}, pl_{g4}\right] \right\}$$

where pl_{g1} is the performance level of the composite indicator, pl_{is} the performance level of the component indicator coming from the influencing situation and $pl_{g2,3,4}$ the performance levels of the sub-goals' indicators.

In case the result is higher than the upper limit (+1) or lower than the lowest limit (-1), the values +1 and -1 are assigned respectively.

Both techniques are rather important for the calculation of KPIs across a hierarchical structure, such the one resulting from the PI structures introduced by Popova and Sharpanskykh [102]. While the 'conversion factors technique' is a quantitative technique resulting in absolute values, supplementing in that way the (simple or trivial) mathematical calculations; the 'range normalization technique' results in performance levels which indicate directly the satisfaction level of the corresponding goals.

Furthermore, Pourshahid et al. [114] have indicated the value of representing graphically the relations between goals and indicators through the notion of 'cognitive fit'. It is stated that "cognitive fit is enhanced when data is presented in a form that fits well with the processes the decision maker uses to make decisions" and, thus, it supports and promotes the decision-making process.

In their method, Pourshahid et al. [114] introduce an extension to the Goal-Oriented Requirement Language (GRL) for supporting a formal evaluation of goals based on KPIs and their aggregate effect. In the proposed model, goals and indicators can be designed in the same graph, while KPIs can be analyzed from different dimensions ('date' and 'location' in the left side of Figure 43), which comprise viewpoints for filtering or aggregating data. For every KPI, its current value and initial satisfaction value are displayed. For an aggregate KPI or a composite indicator (in terms of the BIM approach), the formula for computing its value is contained in the metadata of the model (right side of Figure 43). For the calculation of the satisfaction level of an indicator, the *KPI Performance Level Equation* is used in this paper as well. Hence, the methodology by Pourshahid et al. [114] can be considered a predecessor of the BIM approach by Barone et al. [112], since they provide the basis for incorporating KPI evaluation in a goal graph by the use of formulas and cause-effect relationships among KPIs.

Additionally, Pourshahid et al. [114] provide an iterative framework, called Business Intelligence Decision-Making Framework, for supporting the process of identifying goals and linking them to a decision-model and their corresponding KPIs. As part of their methodology, the authors also consider the risk factor, which is represented as a different concept “contributing” to the composite indicator. The level of contribution causes changes on the threshold value of the indicator.

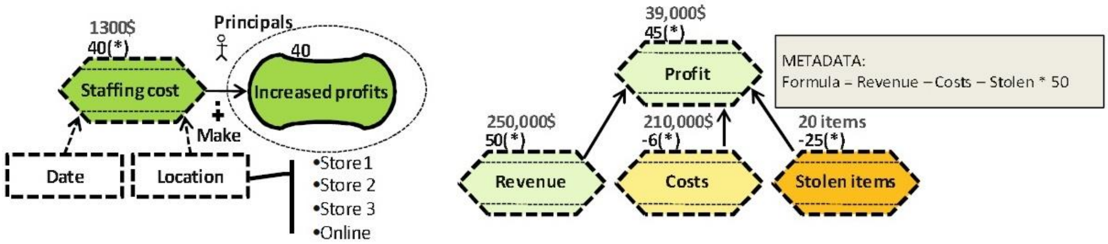


Figure 43 - KPI dimensions, values and metadata, obtained from Pourshahid et al. [114]

To conclude, a cross-check of the relationships identified by Cardoso [103] and the relationships used in the frameworks for evaluating KPIs can provide an overview of the overlap and differences among the methods discussed in this section. The comparison is presented in Table 29.

	correlation	causality	conflict	independence	trade-off	costlier_than	customized	aggregation	computation
Cardoso [103]	x	x	x	x	x	x	x		
Popova and Sharpanskykh [102]	x	x	x					x	
Frank et al. [111]	x (ind.)								x
Barone et al. [112]	x (ind.)	x (ind.)	x (ind.)					x (ind.)	x
Pourshahid et al. [114]		x	x						x
**** ind. is a shortcut for ‘indirectly’***									

Table 29 - Indicator Relations Comparison among Models

By observing Table 29, it can be inferred that for building indicator structures and models for decision-making and analysis of organizational goals, a subset of the identified relations are more useful than others due to their occurrence in more methods. The relation types: ‘independence’, ‘trade-off’, ‘costlier_than’ and ‘customized’ are only discussed by Cardoso [103], which indicates their possibly minor significance for indicator structure modeling. The *causality*, *aggregation* and *computation* relations can be distinguished from the rest, while the *correlation* and *conflict* relations can be considered variations or alternatives of the causality relation.

4.5.4. Performance Evaluation of Goals based on Goal Satisfaction values

As Cardoso has pointed out in her paper [103], goal modeling languages fail to support the relationship between performance indicators and goals. The author proposes two alternatives for assigning KPIs to goals based on the goal levels of abstraction. The first one comprises the assignment of one KPI to every goal in the goal structure (as, for instance, in the BIM approach [112]), while the second one suggests that only leaf (measurable) goals will be associated with

KPIs and the higher-level goals will be assessed based on the evaluation performed in the lower levels. Propagation of goal satisfaction is essential for evaluating the higher-level (more abstract) goals. The criteria, though, for goal satisfaction based on KPIs are poorly researched [103].

Cardoso recognizes the *evaluation type* concept as a solution for specifying the criterion. This concept was proposed by Popova and Sharpanskykh [102] as a method to define goal patterns (as discussed in Section 4.5.3.1). There are three types defined by Popova and Sharpanskykh [102] and considered as useful by Cardoso [103]: achieve/cease, maintain/avoid, maximize/minimize. The correlation between the *evaluation concept* and the *type of scope* dimension of measures (Figure 33) is also stated by Cardoso. The core of the correlation is the *time interval* for the evaluation of goal satisfaction which relates to the restricted set of instances selected for calculating the KPI.

As it is described previously, Barone et al. [112] have used the performance level of KPIs for representing the satisfaction of a KPI and consequently of the corresponding goal. The performance level is the normalized value which results from the calculations for determining the KPI value.

Popova and Sharpanskykh [110] propose an algorithm for propagation of goal satisfaction values, while the goals are defined based on performance indicator expressions and goal patterns, as it is illustrated in the snapshot in Figure 38. The researchers define a set of relations between goals in order to build a goal structure and derive the goals' satisfaction level. The organizational performance is evaluated based on the satisfaction level of the goals not in the design phase but at run time based on collected data [110]. The algorithm's aim is to provide support for "repeated, automated evaluation of performance" [110].

For hard goals, the refinement to lower-level goals is more straightforward as well as the satisfaction of the parent goal. According to Popova and Sharpanskykh [110], when *all* the goals that refine the parent goal are satisfied, the parent goal is also satisfied. The satisfaction level for hard lower-level goals can be directly determined by measuring the values of the related performance indicators [110].

On the other hand, the refinement and satisfaction of a soft goal cannot be decided directly. A soft goal's decomposition is determined through contribution and satisficing relations. The first one indicates the positive relation between two goals, but not the full satisfaction of the parent goal from the child goal; while, a satisficing relation indicates that when the child goal is satisfied, the parent goal is also satisfied. Rules for propagating the satisfaction label of a goal to its parent goals based on the two types of relations are defined in the paper by Popova and Sharpanskykh [110]. Weights can also be assigned to the lower-level goals. The lower-level goals which influence a specific higher-level goal are placed in a list. During the propagation of satisfaction labels, in case of a weighted list of contributing goals, the weighted average is calculated based on a formula of the type: $\frac{\sum_i w_i g_i}{\sum_i w_i}$.

Popova and Sharpanskykh [110] consider five levels of goal satisfaction, which are ranked from 1 to 5: satisficed=satisfied=5, weakly_satisficed=4, undetermined=3, weakly_denied=2, denied = failed = 1. These values are used for determining the satisfaction value of the higher-level goals as it is shown in the algorithm in Figure 44. The method proposed by Popova and Sharpanskykh [110] for the propagation of goal satisfaction values is qualitative, but its benefit is that the determination of the initial satisfaction values of the goals as well as the definition of the goals are based on quantitative performance indicators. The translation of the performance indicator values to goal satisfaction values is not prescribed in the paper, so another method can be used for mapping the values, such as the BIM approach and its normalization technique.

Algorithm for propagating goals satisfaction values:

```
for each depth level i in the goal structure (i = n...0)
    for each goal g in the level i
        if g already has a label, continue
        if g is a hard goal
            identify all refinement lists L_i of g
            g is satisfied if at least one L_i consists of goals which are all satisfied,
            otherwise g is denied
        else g is a soft goal
            identify all influence lists L_i of g
            for each list L_i
                if L_i is an AND-list, then the g's temporary satisfaction value sat_value
                    is determined by the minimal label propagated from the goals in L_i
                else if L_i is a weighted list
                    calculate the sum wtotal of all weights of the goals in L_i
                    calculate the value weighted_label as follows:
                        weighted_label = 0
                            for each goal h in L_i
                                let w be the weight of h in L_i
                                if h satisfices g
                                weighted_label = weighted_label + h.sat_value*w
                            else if h contributes to g
                                if (h.sat_value ≥ 4.5) weighted_label = weighted_label + 4*w
                                if (h.sat_value < 4.5 and h.sat_value ≥ 1.5)
                                                        weighted_label = weighted_label + 3*w
                                if (h.sat_value < 1.5) weighted_label = weighted_label + 2*w
                            weighted_label = weighted_label/wtotal
                    if (weighted_label > sat_value) sat_value = weighted_label
        The satisfaction value of soft goal g is assigned the value of sat_value
```

Figure 44 - Algorithm for Propagating goal satisfaction values, obtained from Popova and Sharpanskykh [110]

4.6. Summary

The Goal Model Analysis techniques explained in this chapter are summarized in Figure 45. They have also been grouped to provide a clearer overview of the methods discussed. In the figure, the methods described in APPENDIX IV are also included.

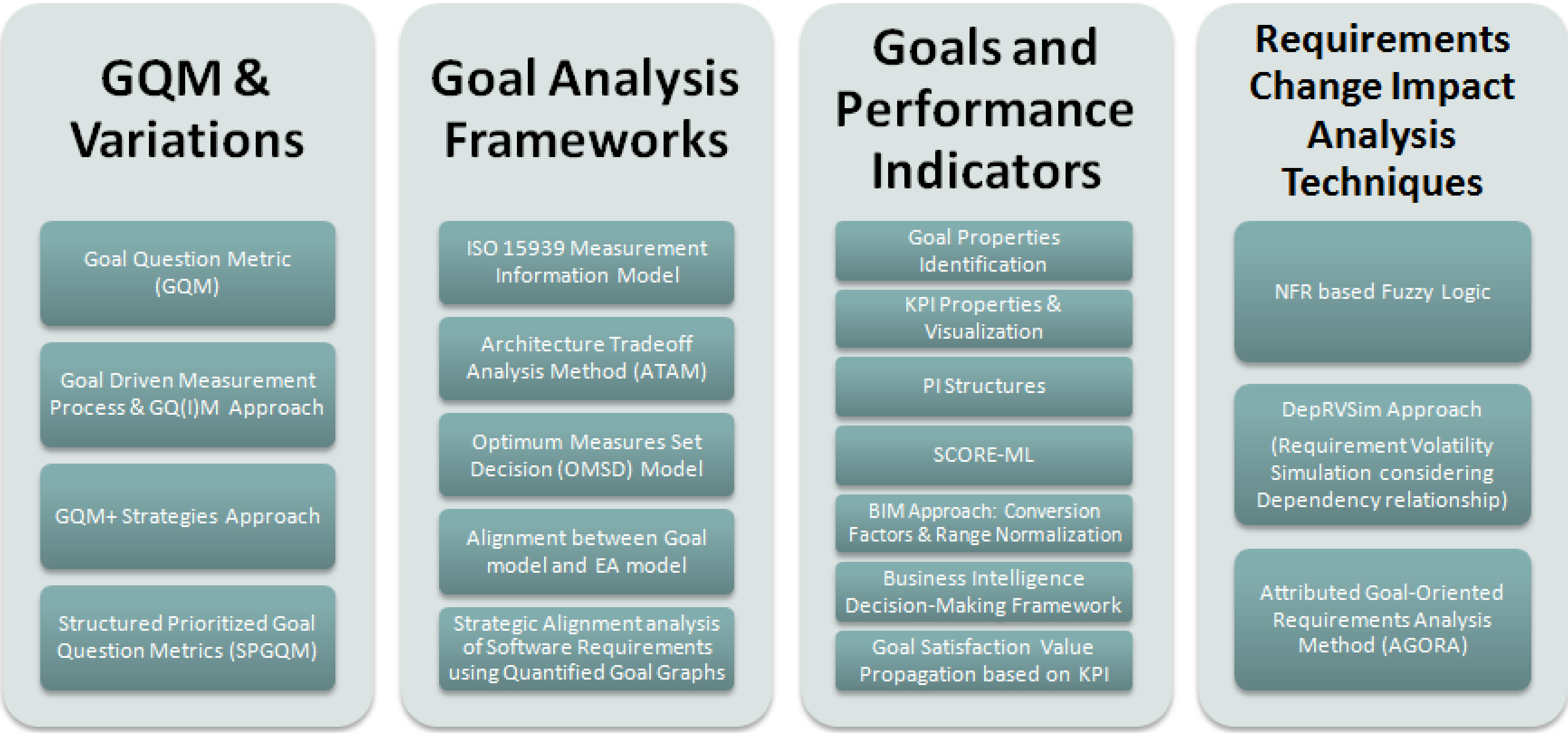


Figure 45 - Goal Model Analysis Techniques Overview

CHAPTER 5

5. An EA-based Goal Quantification Method

In this chapter, the proposed method for quantifying goals based on EA models is described. The design of the method relies on the literature review of Chapters 3 and 4 and comprises the core artifact of the thesis. First, a high-level overview of the method is provided and, then, each step of the method is explained in detail.

5.1. Introducing the EAGQ method

Organizational change is imposed by a variety of reasons which entail uncertainty for the organization. A change can be as small as the addition of another server or as big as the replacement of business processes and services in a world-wide setting. In both cases, a transformation of the existing architecture is necessary and decisions need to be taken for realizing the change while gaining the largest benefits possible. For some transformation projects, more than one alternative solutions may exist; hence, a comparison of the different options in terms of costs, benefits and time would be helpful for determining which is the most appealing solution regarding budget, resources or time available. In other words, the development of a business case is mandatory for evaluating the alternatives. A business case, though, needs to be built on data and measurements which can be analyzed and which can provide more secure and certain decisions.

As it is concluded in Chapters 3 and 4, there are plenty of methods which assist in analyzing the motivational level of change, i.e. the goals, as well as the financial and performance aspect of it, i.e. the architecture level. Since a change is motivated through goals, the expected impact of the transformation, that will take place in the architecture level, should be also visible in the goal level. With the help of ArchiMate®, business goals can become tangible, since the systems, processes, business units or data objects that are associated with the realization of the goal can be traced. Thus, the effectiveness of a solution choice can be mapped to the related goals. Consequently, the estimation of the impact of change in the motivational level in a quantitative manner can be accomplished by relating the quantifications of the EA level to the corresponding goals.

As it has been proposed in Chapter 4, the quantitative analysis in the goal domain can be supported by performance indicator analysis through performance indicator (PI) structures (Figure 46). As for the analysis in the architecture domain, it is supported by architecture analysis models, which determine the objects that are essential for a specific type of analysis and facilitate, in that way, the measurements by isolating the concepts of the architecture model that participate in the measurements.

The characteristics of the PI structures that make them valuable for the quantitative analysis of the goals, are their graphical representation which enables and facilitates the propagation of calculations across the graph as well as the explicit relations among the performance indicators along a path. Furthermore, the one-to-one relationship between an indicator and a goal denotes the direct measurability of the goal, and especially the goal satisfaction, through the indicator's performance.

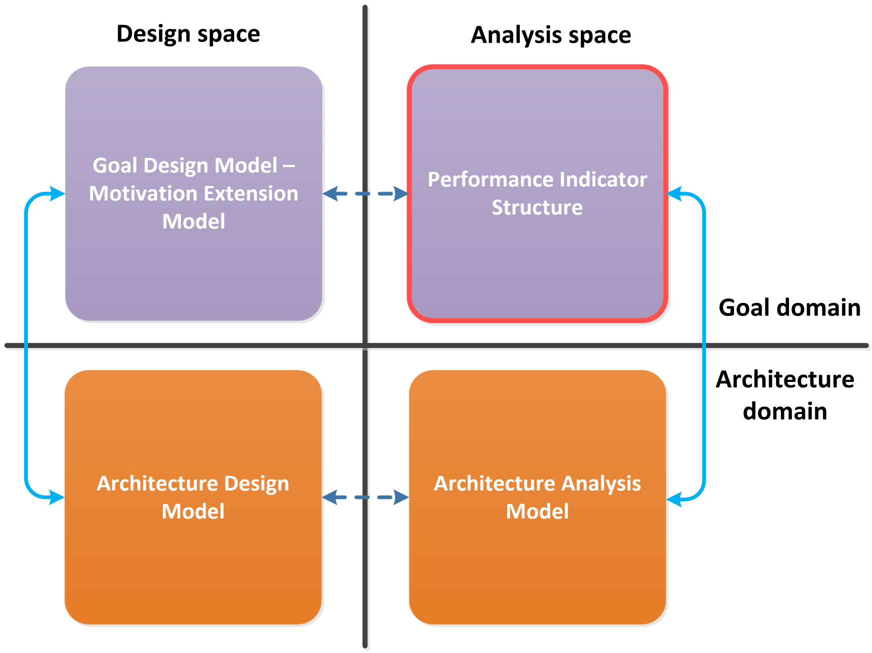


Figure 46 - Design and Analysis space: Focus on Goal domain

The EA-based Goal Quantification (EAGQ) method described in this Chapter aims to combine a variety of techniques and provide guidelines to the user for assigning indicators to goals, relating the indicators with measures derived from the EA concepts, performing the measurements and, then, the analysis of the indicators for identifying the most suitable solution for the motivation model that leads the change. The EAGQ method consists of ten steps, each one of which comprises a set of activities. The order of the steps is designed in such a way to facilitate the application of the EAGQ method and to indicate a logical flow of the actions that need to take place for quantifying the change. The method steps are illustrated in Figure 47, where it can be noticed that they have been grouped according to the Design and Analysis spaces and the Goal and Architecture domains, i.e. 'Goal Model', 'EA Models', 'Indicators and Measures' and 'Data Measurements'.

While the flow is sequential (indicated by the black lines), some steps can be performed in parallel and others can provide input to future steps, as it is explained in the descriptions of the steps and their activities below. The relations and dependencies among the steps of the model are presented in separate graphs in APPENDIX V. Moreover, in APPENDIX V, an overview of the activities of each step can be found.

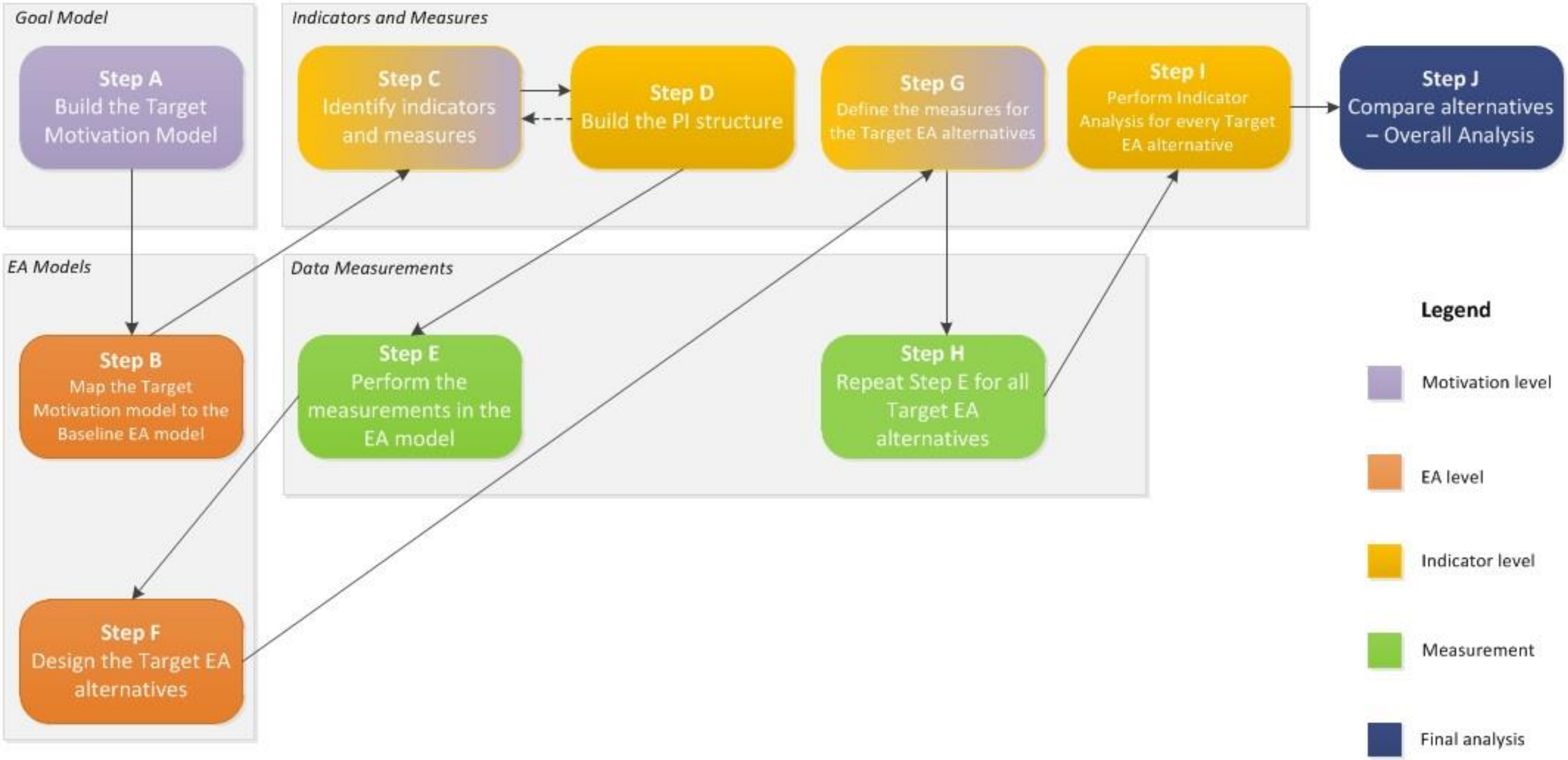


Figure 47 - The EAGQ method

5.1.1. EAGQ Method Terminology – Definitions and Relations

It is important to clarify at this point the terms that are used in the EAGQ method as well as their relations with the ArchiMate® concepts. The new terms introduced are the terms: 'Indicator', 'Measure' and 'Data'. These terms have been used for grouping the steps in the EAGQ method (Figure 47) and they also refer to the two separate domains of Figure 46. Thus, the indicators and measures facilitate the analysis in the goal level, while the data and measurements represent the analysis in the EA level.

The definitions of the terms 'indicator' and 'measure' are based on the SMO definitions provided in Table 23, while the term 'data' refers to input values used in the measurements. In Table 23, the term 'measure' is further refined in 'base measure' and 'derived measure'. In the EAGQ method, this differentiation between types of measures is not going to be used. Additionally, the SMO distinguishes 'measurement approaches' as measurement methods and functions, which correspond accordingly to the base and derived measures. Both differentiations are considered confusing. The aim of the EAGQ method is to introduce the necessary concepts for performing the measurement in the EA level and the analysis in the goal level, while avoiding additional complexity. Therefore, the terminology used in the method is presented in Table 30. The definitions for the terms 'measure' and 'measurement approach' are derived from the definitions in Table 23. Additionally, as far as the 'analysis model' is concerned, the definition provided in Table 30 is a result of the definition in Table 23, but it also aligns with the definition of the 'analysis model' discussed by Jansen et al. [36] as part of the Quantitative Analysis Process (Table 8). The diagram in Figure 48 provides the relations between the terms.

Term	Definition
Measure	A measure of an attribute which is a result of a measurement approach. A measure can be also calculated by using other measures.
Indicator	A measure that is derived from other measures using an analysis model.
Measurement approach	The measurement method used for calculating the value of a measure. It comprises a set of operations as well as an algorithm or a calculation formula. Defining the measurement approach also includes the measurement scale.
Analysis model	An analysis model is a generalization of a measurement approach. The analysis model includes a measurement approach for calculating the value of an indicator and decision criteria for performing the analysis of the calculated value.

Table 30 - EAGQ method Terms and Definitions

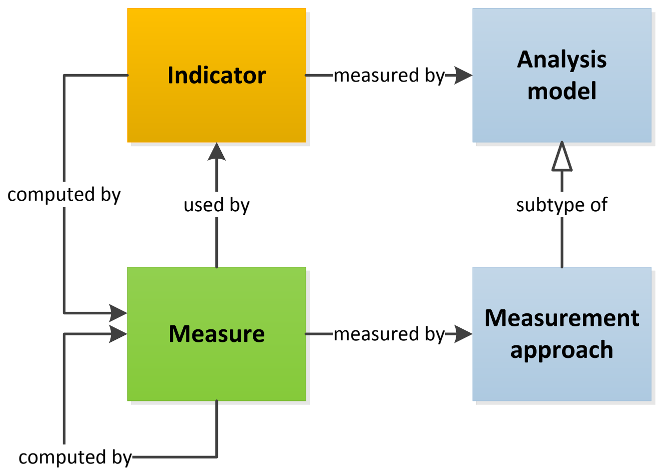


Figure 48 - Relations between terms in the EAGQ method

A proposal of the correlation between concepts of the goal model and the EA model and the terms ‘Indicator’, ‘Measure’ and ‘Data’ is presented in Figure 49. The indicators, as it has been explained, correspond to ‘goals’. The measures, though, correspond to ‘requirements’ and derive information from EA concepts.

The term ‘requirement’ in ArchiMate 2.1 Specification [17] is defined as “a statement of need that must be realized by a system”. The fact that requirements provide more precise information regarding the change in stake and the needs that the target EA should fulfill, makes it an ideal concept for representing the measurement needs imposed by the goals. Thus, requirements serve as intermediaries for translating the goals into system needs, while the measures serve as the intermediaries for translating the indicators into measurement needs. The measures corresponding to requirements, though, need to be derived from the EA model. The measures in the requirement-level are usually more complex measures, which are calculated from other simpler measures, which are called component measures for clarifying the relationship between them. Hence, the requirement-level provides the point where the aggregation of the measurements occurs.

The measurement approaches of the measures are the focus of the ‘Data Measurements’ box in Figure 47. The measurements are performed in the EA model and the term ‘data’ corresponds to any kind of concept in the EA model which realizes a specific requirement. To put it differently, as the EA concepts realize the requirements, the data are the input values used for computing the measures related to the requirements. The source of the data lies on the concepts participating in the measurements.

In the next sections, every step of the EAGQ method and its corresponding activities are presented and analyzed. Overall, the method aims to assist and guide the quantitative analysis of change in order to justify its contribution to fulfilling the goals. The quantitative analysis in the goal domain is achieved by taking into consideration the current situation, as represented by the Baseline EA model, and comparing it with estimations of the expected results that the transformation will cause. The comparisons are performed based on data populations of the EA models and quantifications on these data.

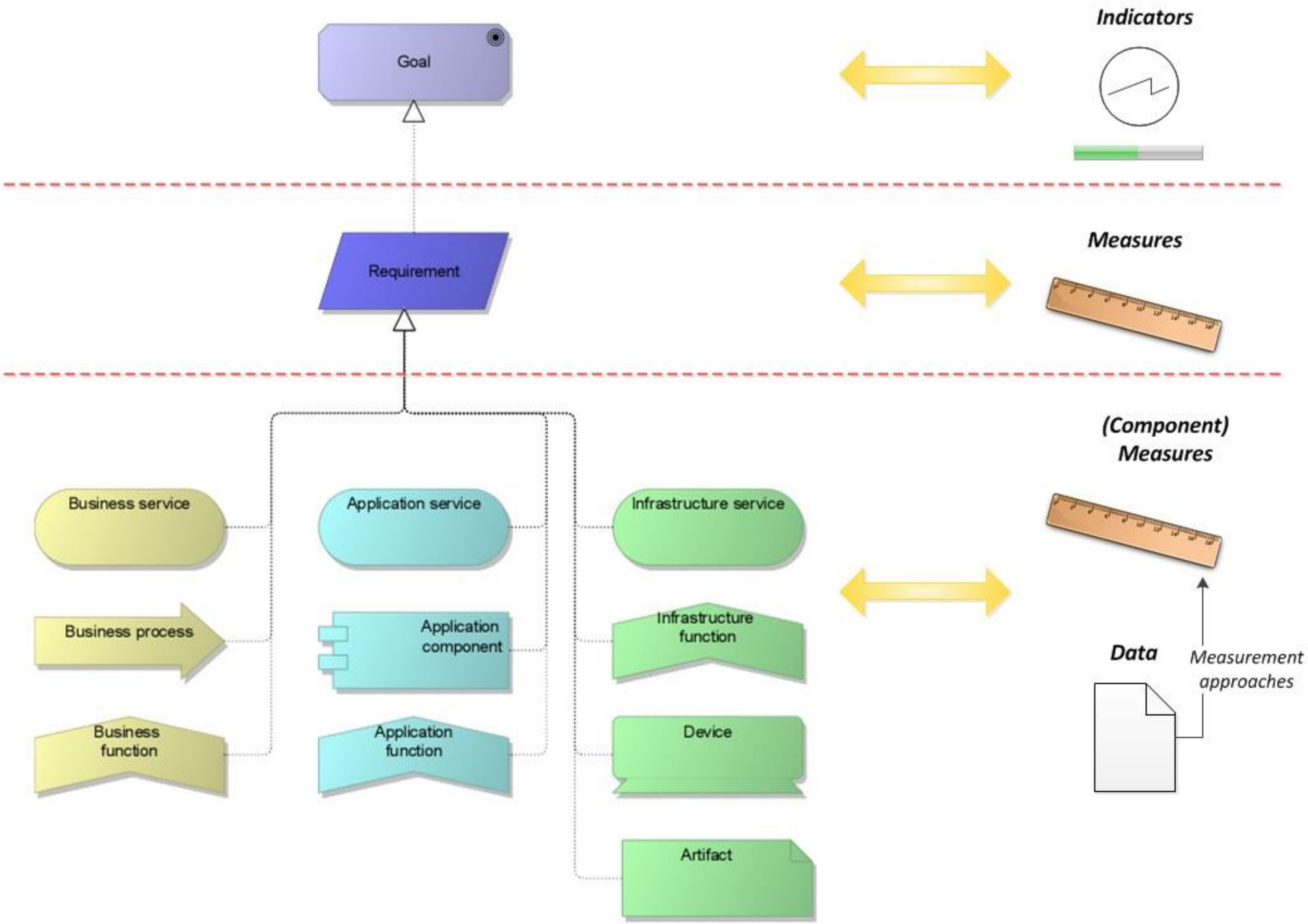
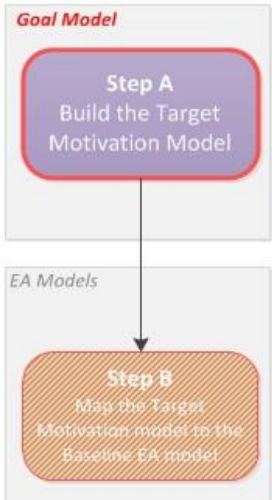


Figure 49 - Correspondence of Indicators, Measures and Data with ArchiMate® concepts

5.2. Explaining the Steps and Activities of the EAGQ Method

Step A: Build the Target Motivation Model



The need for change and migration to a new architecture is expressed through the motivation model. Since the motivation model, i.e. the goal model, serves as the guideline for leading the change, the first step of the model aims to build the Goal Model and determine the specific attributes of the goals. These two concerns comprise the activities of Step A. An overview of the step including its activities, input and output data are illustrated in Figure 50.

The notation in Figure 50 (and all the similar figures in the next steps) follows the Business Process Model and Notation (BPMN) language. Each step is considered a separate process with its own tasks (blue boxes). The input data for each task, or activity in terms of the EAGQ method terminology, are represented with the empty-arrow data file image, while the output of each activity has a filled arrow to indicate the difference. The output (filled-arrow) data files of an activity are used as input to following steps or, internally in a step, to another activity.

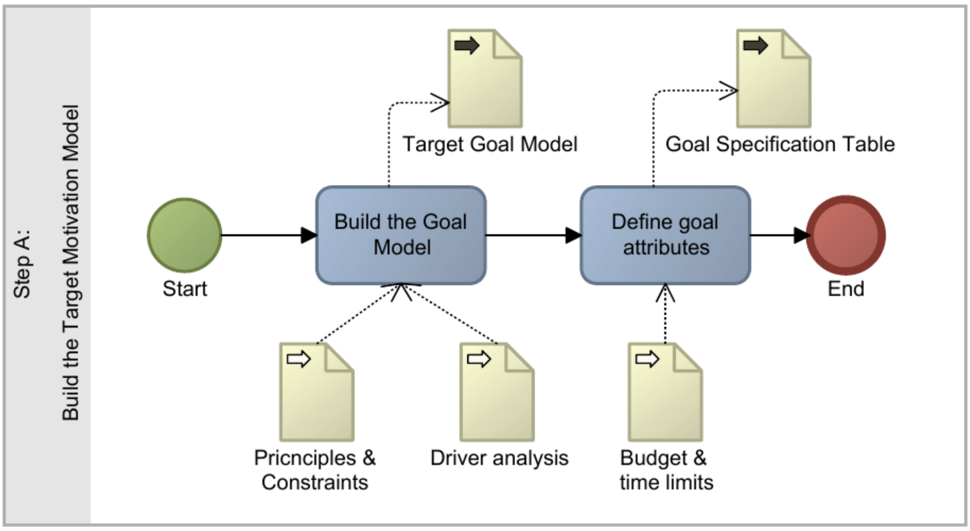


Figure 50 - Step A: Activities, Input & Output data

Activity A1. Build the Goal Model

The purpose of this activity is to build the target motivation extension model. Due to the focus on quantification analysis, the concepts of the ArchiMate® Motivation Extension used in the method are the goals and requirements. These concepts can express the need for change and can have direct influence on the decisions regarding the transformation of the baseline architecture. While other concepts can exist in a motivation model, such as 'drivers', 'assessments', 'constraints' and 'principles', the outcomes of measurement and analysis originating from the architecture cannot be assigned to or inherited by them. The only exception could the concept of 'assessment'. It is defined as the "outcome of some analysis of some driver" in the ArchiMate 2.1 Specification [17], which restricts its ability of representing the outcome of the analysis performed from the goals' perspective. A generalization of that definition could enable the use of the 'assessment' as the concept for representing the indicators in the PI structure.

As for the relationships among goals, the ArchiMate 2.1 Specification [17] defines four types of relationships: aggregation, realization, influence and association. Since the 'association' relation is the weakest relation and does not specify how two goals relate with each other, it cannot be used for goals that are interdependent for their accomplishment. The main assumption in this activity for designing the goal model is that the goals, as they are further refined, express a computational relationship with their lower-level goals which will facilitate the determination of indicators later on. Hence, the 'aggregation' and 'realization' relations are the more concrete ones, while the 'influence' relation may be questionable, since it indicates a kind of correlation between the goals. This issue is also discussed and explained in more detail in Step D where the design of the indicator structure takes place.

Since the Target Motivation model denotes the to-be situation, it is the outcome of a higher level analysis performed by various stakeholders, who mainly belong to the Boardroom category of Table 1 (for examples see Table 31). Thus, the input for designing the goal model is the assessment of a set of drivers. 'Constraints' and 'principles' can guide the decisions while designing the goal model, but as it is explained, they will not be taken into consideration during the quantitative analysis. The outcome of the drivers' analysis usually results in high-level business goals which need to be decomposed in lower-level measurable goals. Techniques that can support this procedure are:

- SPGQM approach and particularly the Goals Definition Phase
- GQ(I)M: Steps 1-4 – Refining Business goals to Measurable goals (Table 20)
- GQM$^+$ Strategies approach for decomposing goals through strategies

For more details regarding the techniques, the reader can refer to Chapter 4. Except from the Boardroom stakeholders, stakeholders from the IT Management group can participate or be consulted especially for the determination of the Requirements.

To conclude, the ArchiMate® motivational concepts and relationships that will be used for building the target motivation model – goal model – which will guide the transformation, should be the following:

Concepts	**Relationships**
• Goal • Requirement	• Aggregation • Realization • Influence

Table 31 summarizes the explanation of this activity.

<table>
<tr><th colspan="2">Activity A1. Build the Goal Model</th></tr>
<tr><td>Goal</td><td>To build the Goal Model which will guide the transformation of the architecture</td></tr>
<tr><td>Stakeholders</td><td>Boardroom, for instance from Table 1:
CEO, CFO, COO, CIO
Business Change Manager
Business project manager
Operational business manager
IT Management, for instance:
IT change manager and/or Information analyst
Software designer/architect
Infrastructure engineer and/or administrator</td></tr>
<tr><td>Assumption</td><td>The refinement and decomposition of goals into lower-level goals expresses an explicit or implicit computational relationship among them.</td></tr>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>Drivers analysis
Assessment
Constraints
Principles</td><td>Target Goal Model
Target Requirements</td></tr>
<tr><th colspan="2">Techniques</th></tr>
<tr><td colspan="2">SPGQM: Goals Definition Phase
GQ(I)M: Steps 1-4 – Refining Business goals to Measurable goals (Table 20)
GQM+ Strategies approach for decomposing goals through strategies
[Chapter 4, Figure 45]</td></tr>
</table>

Table 31 - Overview of Activity A1

Activity A2. Define goal attributes

As it has been described in Sections 4.5.1, 4.5.3 and 4.5.4, there are a set of goal attributes that, when defined, can assist the measurability and explicitness of the goal. The second activity of Step A aims to identify the following goal attributes for each goal in order to facilitate their quantification:

- **Degree of measurability:** soft, hard
  Identifying a goal as 'soft' has a twofold meaning; either that the goal is not quantifiable, but it can be characterized as qualitative; or that the goal has not been properly refined yet, which would lead back to Activity 1 for further decomposition. When a goal's evaluation is qualitative, e.g. customer satisfaction, it is highly unlikely that this goal can contribute in the forthcoming calculations. In case an 'influence' relation links that goal with another 'hard' goal, it is acceptable to exclude it from the computations. Such goals could be used, though, for supporting or diminish the outcome of the indicator analysis. The way that soft goal may be used is up to the stakeholders.
- **Direction or evaluation type or goal pattern:**
    - Achieve
    - Avoid
    - Cease
    - Maximize (increase)
    - Maintain
    - Minimize (decrease)

The determination of the evaluation type of the goal indicates its intention and it is derived from the name of the goal.

- **Target value**
  Depending on the expected analysis on the goal level, the target value of the goal can be expressed in percentages (e.g. Increase profit by 15%) or in absolute numerical values (e.g. Increase profit by 20.000€). This piece of information will be determinant for the way that the indicators will be defined.
- **Timeframe or Deadline** for goal achievement
  The timeframe, that the goal should be achieved in, indicates the urgency of the goal. Additionally, it can be used for comparing possible alternative solutions and assessing their value towards the goal. The way to assess the value of the alternative is discussed in Step C.

In Section 4.5.1 more attributes are identified. Table 32 provides the reasons for not using these attributes in the EAGQ method.

Attribute	Reason
Rationale	This attribute does not contribute directly in the quantifiable analysis, but it is essential to be determined during Activity 1, when the goal model is built.
Stakeholder	It is a supportive attribute which can determined for each goal, since the stakeholder could facilitate or monitor the measurement and the progress of the goal. The focus, though, in this model is to use the attributes that are defined for each goal.
Restrictions and Implications	Both attributes do not relate directly with the quantification analysis.
Interrelations	This attribute is already covered by the goal graph and its development with the use of ArchiMate® Motivation Extension.
Focus area	Used in Step C and Step G.

Table 32 - Goal attributes and why excluding a subset of them

Finally, it is convenient to state the goal's tree level because it denotes each goal's distance from the requirements. This task is related with the fact that the indicators, as it is explained in Step C, are defined firstly for the goals in the lowest-level of the graph. Thus, the goals directly realized by the requirements are assigned the level-value 0. In case asymmetries exist in the tree, to avoid various level-values of the highest level goal of a sub-graph, two solutions are possible:

1. Assign values only to the lowest-level goals, i.e. 0 to the goals directly linked to the requirements. This contributes to simplicity and less complexity.
2. Organize the goals in levels and for the paths that do not have a goal in a specific level, leave this level blank. Figure 51 shows this solution.
   In the ArchiMate® notation, the requirement and goal concepts have the same color. For convenience and better identification of the requirements in the goal model, the requirements have a darker purple color.

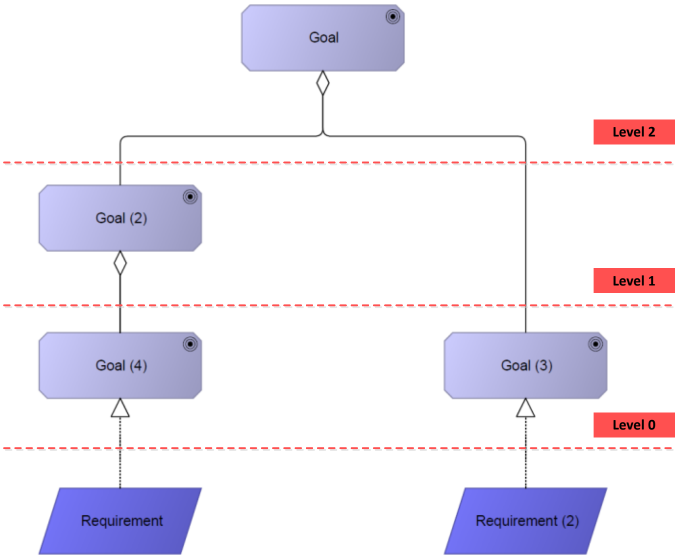


Figure 51 - Assignment of level-values to Goals and Requirements

To present the identification of the goal attributes, the Goal Specification Table is used, where each goal is documented. The template of the Goal Specification Table is provided in APPENDIX VI. Table 33 provides an example of the Goal Specification Table, where the first goal needs to be further refined in order to be realized by a requirement, while the second can be realized by a requirement directly. A requirement that can realize the 'Improve delivery time' goal is the 'Integration of Sales & Order and Inventory management systems'.

Goal Specification Table					
	Attributes				
	Level	Measurability	Evaluation type	Target value	Timeframe or Deadline
Goal	Tree level (0 – n)	(S)oft or (H)ard	(A)chieve (C)ease (M)aintain A(v)oid (I)ncrease (D)ecrease		
Achieve lower price	1	H	A	265€ per item	14 months
Improve delivery time	0	H	D	Decrease by 2 working days within Netherlands	12 months

Table 33 - Goal Specification Table

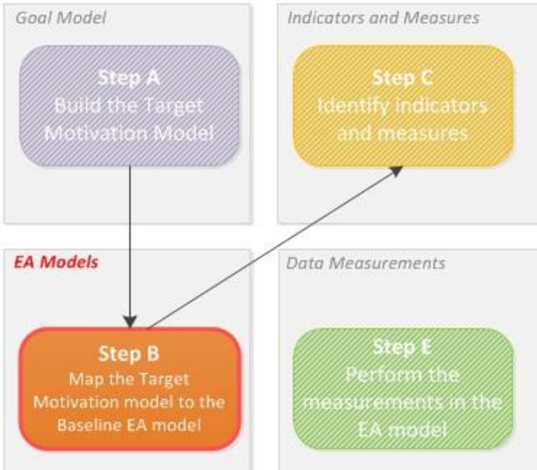
The overview of Activity A2 is provided in the next table (Table 34).

<table>
<tr><th colspan="2">Activity A2. Define goal attributes</th></tr>
<tr><td>Goal</td><td>To determine the specific attributes of the goals</td></tr>
<tr><td>Stakeholders</td><td>Boardroom, for instance:
• CEO, CFO, COO, CIO
• Business Change Manager
• Business project manager
• Operational business manager</td></tr>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>• Budget
• Time limits</td><td>• Goal Specification Table</td></tr>
<tr><th colspan="2">Techniques</th></tr>
<tr><td colspan="2">Techniques that can support this activity are the following:
• Goal Properties Identification</td></tr>
</table>

Table 34 - Overview of Activity A2

Step B: Map the Target Goal Model to the Baseline EA model

When the target goal model is designed, the next step comprises relating the goals to the Baseline EA model. This task is necessary for enabling the assessment of the potential target architectures. The purpose of this step is to assess the baseline architecture against the target goals. By knowing how the baseline is performing, it can serve as the basis for analyzing the expected performance, costs or other attributes in interest of the target architectures.



The connection between the goal model and the architecture model is achieved through the requirements 'layer'. The requirements can be directly realized by concepts in the architecture, can be partly realized or not fulfilled at all. Thus, a first estimation and identification of the gaps between the current situation and the expected future functionality can be determined. This pre-estimation of gaps supports the design of the target EA alternatives as well. The value and contribution of Step B throughout the whole EAGQ method is illustrated in APPENDIX V (Figure 121). Step B includes three activities, as it is illustrated in Figure 52.

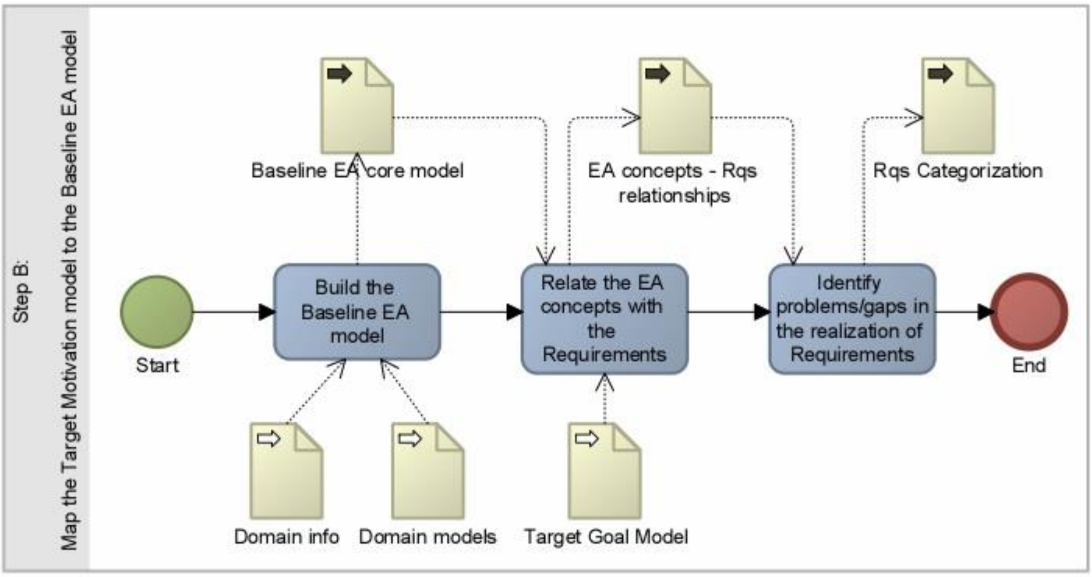


Figure 52 - Step B: Activities, Input & Output data

The main deliverables of Step B are the Baseline EA model, the relations of the EA model with the Goal model and the categorization of the requirements depending on their degree of realization by the current architecture (filled-arrow data files in Figure 52.

Activity B1. Build the Baseline EA model

The main goal of the EAGQ method is to support and guide the indicator analysis which substitutes the quantitative goal analysis. The analysis entails the comparison of the possible architecture alternatives regarding their performance against the baseline architecture. In this way, the benefits and the progress that the target architecture offers to the organization can be calculated. To be able to make the comparison between the architectures, designing the baseline EA model is the initial step. The design of the model is not only the input for the next activity, as it is illustrated in Figure 52, but also the basis for determining the measures in Step C and deriving the EA analysis model in Step E (APPENDIX V, Figure 121).

Building the architectures requires the involvement of various stakeholders from all three domains, namely business, application and infrastructure domains. Thus, the responsibility is upon the IT Management stakeholders' group for designing the EA model and verifying that all essential information for future analysis are represented in the model. The latter comprises one of the main assumptions of this activity. It is also assumed that the core information for making the model understandable and useful for the measurements and the analysis can be found in the organization. Relevant information may also include partial models referring to separate domains. Gathering the information or the domain models can be achieved through interviews with the domain specialists and the IT Management stakeholders. The overview of the activity is provided in Table 35.

Activity B1. Build the Baseline EA Model	
Goal	To build the Baseline EA model for facilitating goal model and target architecture evaluation.
Stakeholders	*IT Management*, for instance: • Division Information Officer • Data administrator • Software development project manager • Software designer/architect • Application management and/or administrator • Platform manager • Data center management • Infrastructure engineer and/or administrator
Assumptions	• The Baseline EA model is possible to be designed. • All the essential information is available. • The output model of this activity provides enough information or a good basis for further analysis.
Input	**Output**
• If Baseline EA model is *available*, then this activity can be skipped. • If the architecture is *not available*, then information or partial models per domain should be provided as input.	• Baseline EA model
Techniques	
Interviews with specialists in each domain and IT Management stakeholders	

Table 35 - Overview of Activity B1

Activity B2. Relate the Baseline EA concepts with the Requirements

As it has been already stated, 'requirements' function as the intermediary between the goal model and the EA model. Therefore, mapping Baseline EA concepts with the target requirements indicates how the baseline architecture contributes to the target goals. It is expected that not all requirements can be realized by the baseline architecture. Thus, two types of relationships are to be used for representing the connection between requirements and EA concepts:

- **'Realization'** relation: If there is a direct realization of a requirement by one or more concepts, this relation type is going to be used.
- **'Association'** relationship: This relation type will be used to indicate that a concept of the architecture contributes to a requirement, despite the fact that the concept does not fulfill the requirement completely.

The meaning of distinguishing between the two types of relations and the advantages it offers are explained in Activity B3, which accepts as an input the relations determined in this activity.

Assigning core concepts to requirements is a task of IT Management stakeholders and specialists of the domains who can determine whether the needs expressed by the requirements are met by the current architecture. Interviews, discussions among specialists as well as reference to documentation of requirements, systems, processes, services, databases, etc. are techniques that facilitate the mapping. Assuming that the people involved in the activity have the knowledge and experience to perform the activity, the outcome of the task also depends, firstly, on the availability of architecture views providing more or less detail for enabling the relation between requirements and core concepts; and, secondly, on the availability of information for designing the views. Finally, when a requirement cannot be matched to any core concept, there are two possibilities; either the concept that should realize the requirement does not exist or it cannot be found due to complexity reasons or insufficient documentation. In such a situation, the user should assume that the requirement is not realized and that an action should be taken in the target architecture to fulfill it. All three possibilities regarding the mapping of EA concepts to requirements are depicted in Figure 53.

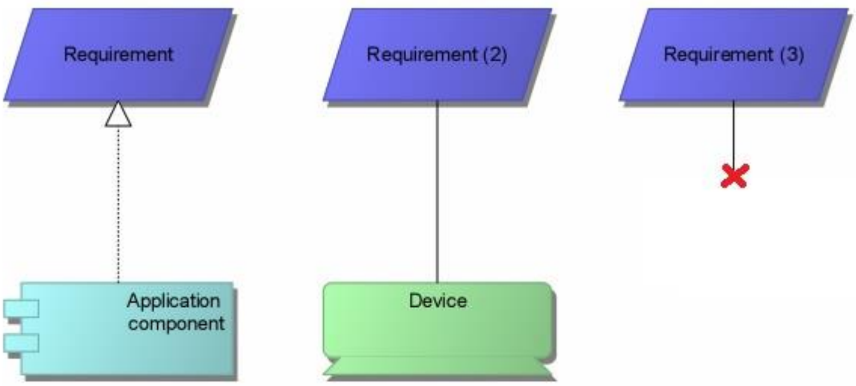


Figure 53 - Requirements and EA concepts relations in the EAGQ method

<table>
<tr><th colspan="2">Activity B2. Relate the Baseline EA concepts with the Requirements</th></tr>
<tr><td>Goal</td><td>To identify and connect the Baseline EA concepts which realize or are related to the target requirements.</td></tr>
<tr><td>Stakeholders</td><td>IT Management, for instance:
• IT change manager
• Information analyst
• Data administrator
• Software designer/architect
• Application management and/or administrator
• Platform manager
• Data center management
• Infrastructure engineer and/or administrator</td></tr>
<tr><td>Assumptions</td><td>• The views and the concepts included in the Baseline EA model are sufficient for relating the requirements with the concepts
• There is the possibility to design other views (more or less detailed) for identifying potential existing realization relationships
• For the requirements that no match can be found, assume that currently there is no realization.</td></tr>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>• Target motivation model
• Baseline EA model</td><td>• Realization and association relationships among Requirements and EA concepts</td></tr>
<tr><th colspan="2">Techniques</th></tr>
<tr><td colspan="2">• Interviews with stakeholders and specialists
• Discussions among specialists
• Cross-checking of or reference to documentation of requirements, systems, processes, services, databases and data objects, etc.</td></tr>
</table>

Table 36 - Overview of Activity B2

Activity B3. Identify problems/gaps in the realization of the Requirements

Gap analysis is a controversial or abstract term which can be applied in various situations. In the EAGQ method, the gap analysis is performed by determining which of the requirements are realized by the baseline architecture, which of them are partly addressed and which are not fulfilled at all. Categorizing the requirements in these three categories (Table 37), the existence of gaps or problems can be estimated and the determination of target architecture alternatives can be facilitated (see dependency relations among steps in APPENDIX V, Figure 121).

Relation type	Requirement realization-level	Gap estimation
Realized	**Covered**	• No potential gap • Requirement underperforming
Associated	**Partly covered**	Potential gap: • Requirement underperforming • Requirement partly fulfilled
No relation	**Not covered**	Gap existence: • Requirement not realized at all

Table 37 - Requirement Categorization for Gap estimation

In Table 37, one of the three requirement realization-levels is called 'partly covered'. An example of such a requirement is, for instance, when there is a service realizing the requirement, but the system that realizes the service does not fulfill the requirement's need. Consequently, there is a partial coverage of the requirement (Figure 54). In case, a requirement is realized, there is still, though, the possibility that it can be underperforming, since more concepts and factors can influence it. While such a situation may not be obvious, it could be revealed after the transformation has occurred. Figure 53 illustrates all three categories.

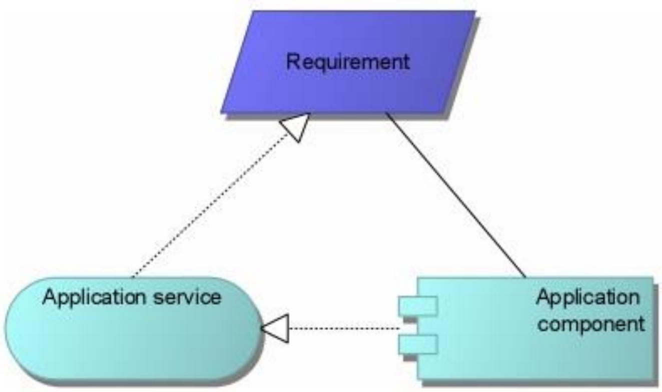


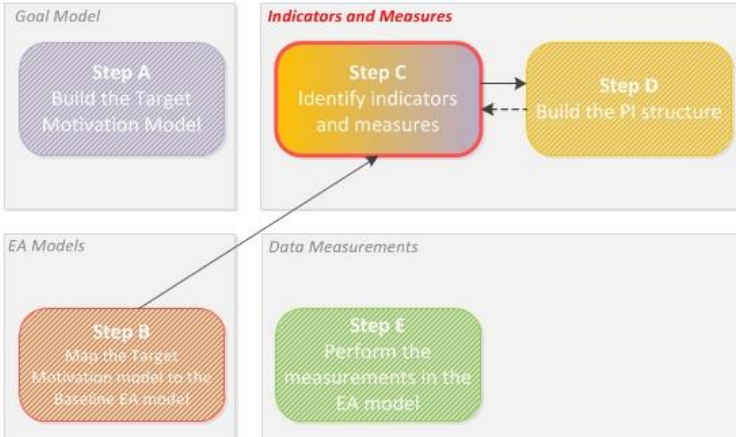
Figure 54 - Partly covered requirement example

The advantage of this activity resides in the fact that it can assist budget management of the change project. Due to the categorization of requirements, a kind of prioritization is achieved which can guide the budget distribution. By identifying requirements that are already covered, even if they may not perform perfectly, it serves as an indication that the focus should be placed upon the rest and, thus, the budget should be invested to cover other needs. This is very important when there is risk of running out of budget in a project or when the transformation is split in steps and the prioritization of the steps should be decided.

<table>
<tr><th colspan="3">Activity B3. Identify the problems/gaps in the realization of the Requirements</th></tr>
<tr><td>Goal</td><td colspan="2">To facilitate the design of target alternatives by denoting the realization-level of the requirements.</td></tr>
<tr><td>Stakeholders</td><td colspan="2">IT Management: possibly the same stakeholders that were involved in the previous activity.</td></tr>
<tr><td>Assumption</td><td colspan="2">Sufficient information is provided by the previous step to support the assessment of the requirements' realization.</td></tr>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td colspan="2">Relationships among requirements and Baseline EA concepts (output of Activity 2)</td><td>Based on the realization estimations, categorization of requirements as:
Covered
Partly covered
Not covered</td></tr>
<tr><th colspan="3">Techniques</th></tr>
<tr><td colspan="3">Discussions among specialists
Interviews with stakeholders and specialists</td></tr>
</table>

Table 38 - Overview of Activity B3

Step C: Identify indicators and measures



To be able to perform the analysis in the goal domain, indicators must be assigned to each goal. The indicators represent the way the goals should be quantified as well as the way their progress should be interpreted. Additionally, the presentation of the analysis outcome through visualizations is a key concept for demonstrating the value of the analysis process. Therefore, this step includes the definition of the visualizations associated with each indicator. Finally, the goal specification is also completed, as it is mentioned in Table 32. Consequently, Step C includes four activities which are analyzed below. Figure 55 presents the overview of the step.

Every activity in Step C has its own output, but the main deliverables that comprise the essence of the step are the Indicator Specification tables and the Measure Specification tables. These tables are going to facilitate the design of the PI Structure in Step D as well as the analysis of the indicator and the goal performance in Step I. Further value of Step C and its influence on other steps of the EAGQ method are discussed in APPENDIX V, Figure 124.

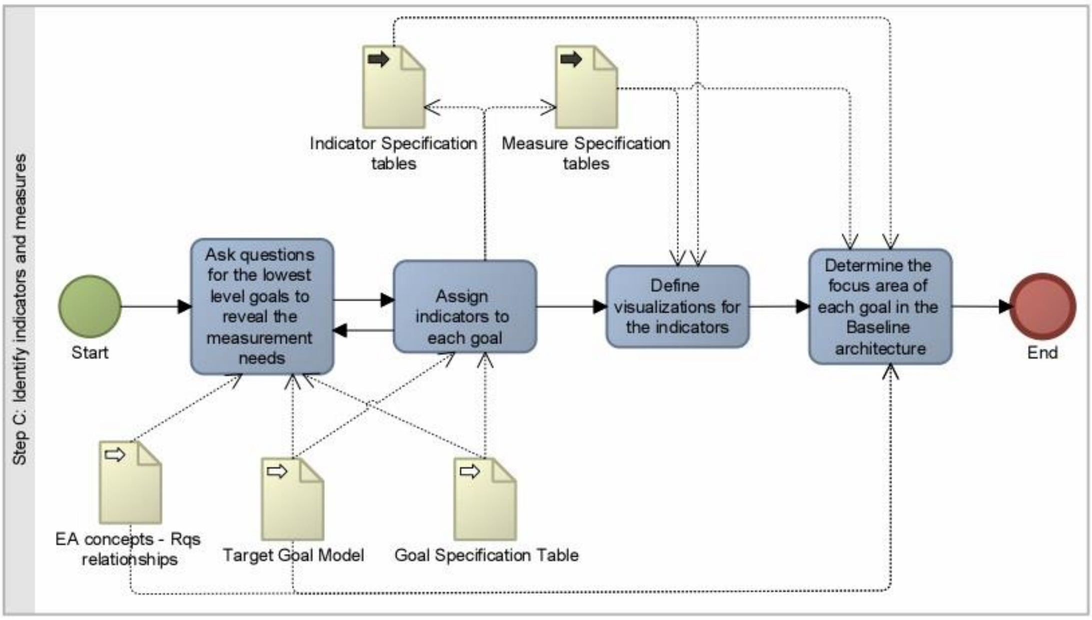


Figure 55 - Step C: Activities, Input & Output data

Activity C1. Ask questions for the lowest-level goals to reveal the measurement needs

The primary task of this activity is to ask questions to reveal the measurement needs. In this activity the GQM or GQ(I)M approach is applied for the lowest-level goals, i.e. the goals with level-value equal to zero. While asking questions for every goal in the goal model is an option, such an activity would be too time-consuming for large goal models. Thus, the focus is upon the goals which are directly realized by the requirements.

As it is described in Section 4.1.1 and in Table 19, the *questions* aim to characterize the *elements/objects* to be measured with respect to an *attribute/issue* and a purpose, i.e. the evaluation type of the goal specified in Activity A2. A goal's issue may not reveal the measurement needs of higher level-goals; hence, when asking questions, considering both higher-level goals and the requirements, which realize the goal, is necessary. The questions have to address the goal under analysis, but they should also be in line with the focus of the higher-level goals belonging in the same path with the goal (e.g. focus on cost, performance, etc.). Thus, a goal should not be considered in isolation from the rest of the model.

The *elements, objects* or *entities*, that the attributes refer to, comprise EA concepts. For better support to Step C, and especially Activities C1 and C2, Activity E1 ("Build the EA Analysis model for every lowest level goal and its respective requirements") can be performed in parallel (Figure 56). In that way, the visualization of the sub-graph of the architecture where the data can be derived from, i.e. the EA analysis model, will provide help for the specification of the entities that are related to a goal and direct the questions referring to the entities.

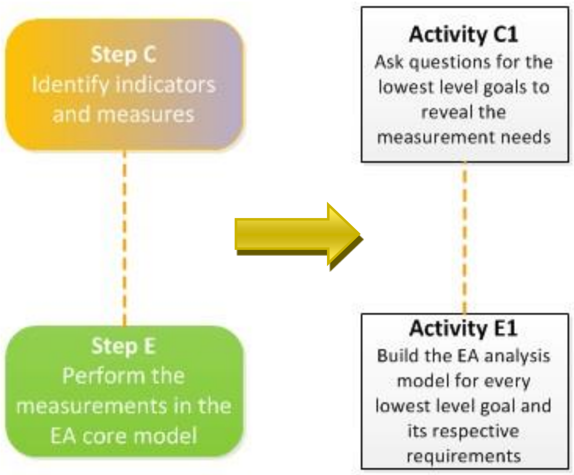


Figure 56 - Parallel application of activities in Steps C and E

Additionally, knowing the EA analysis model will provide better guidance for deciding on what measurements are possible, identifying the most suitable measures in the requirement level and determining the required data in the architecture level. These tasks are part of Activity C2, where the indicators and the corresponding measures are defined. Since Activities C1 and C2 are interrelated, it is expected that an iteration of this activity may be necessary while the indicators and measures are being determined (Figure 57).

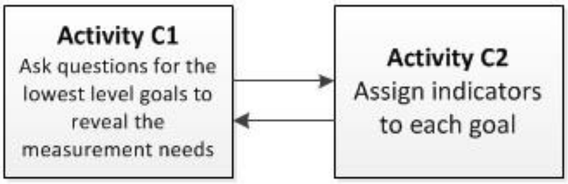


Figure 57 - Step C: Iteration between Activities C1 and C2

Another advantage of asking questions for discovering measures for goals is the fact that the *level of detail* of the data and measurements to be performed for characterizing the selected attributes can also be determined. For instance, Table 75 presents a set of cost categories, cost factors and cost elements. While for one cost analysis defining the acquisition costs in terms of more generic cost factor groups (e.g. hardware costs, software costs, etc.) would be sufficient, for another case the determination of the exact elements that comprise the factors would be crucial. Thus, the questions can direct the expected level of detail of the measurements.

Example questions can be found in Section Section 4.1.1 (Operational level) as well as in the papers by Basili et al. [80] and Park et al. [81]. A demonstration of the model is provided in Chapter 6, where more example questions can also be found.

An important prerequisite for the GQM and GQ(I)M methods is the addressed goals to be measurable. This requirement is already satisfied due to the ArchiMate® modeling principles and the application of Step A.

To sum up, the overview of the activity is presented in Table 39, where it can be noticed that the input of the activity is, apart from the output of Step B, the Target goal model designed in Activity A1 and the goal level-values defined in the Goal Specification Table (Activity A2).

<table>
<tr><th colspan="2">Activity C1. Ask questions for the lowest-level goals to reveal the measurement needs</th></tr>
<tr><td>Goal</td><td>To determine the questions which will lead the measurements identification for the goal analysis</td></tr>
<tr><td>Stakeholders</td><td>Boardroom and IT Management</td></tr>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>• Target Goal model
• Goal level-values from the Goal Specification table
• Realization and association relationships among Requirements and EA core concepts</td><td>• List of questions for every goal with level-value equal to 0</td></tr>
<tr><th colspan="2">Techniques</th></tr>
<tr><td colspan="2">• GQM
• GQ(I)M: Step 6 - Identify quantifiable questions [and the indicators that will be used for achieving the measurement goals]
• SPGQM: Questions Definition Phase</td></tr>
</table>

Table 39 - Overview of Activity C1

Activity C2. Assign indicators to each goal

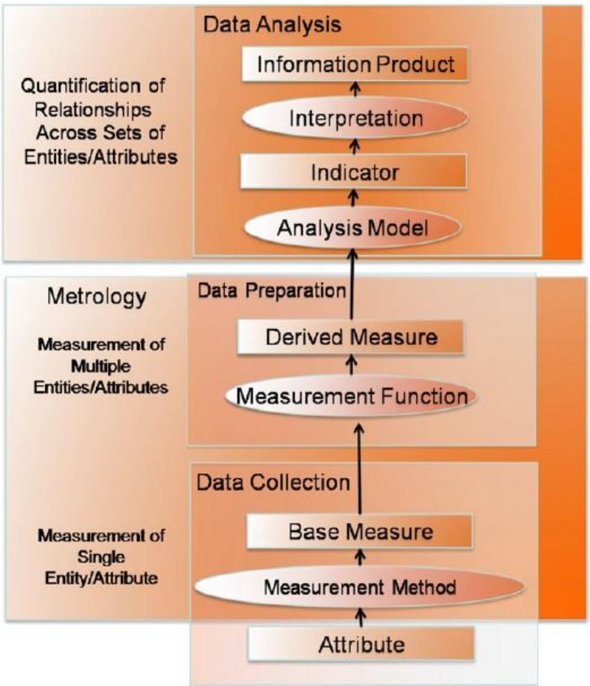
As soon as the questions have been determined, the next activity is to answer the questions and assign indicators to every goal in the goal model. The questions in Activity C1 have been mostly defined for the level-value 0 goals, which indicates that these are the goals to start the indicator assignment process.

Activity C2 is the core of Step C and provides the background for performing the quantitative analysis in the next steps. The purpose of Activity C2 is to determine the indicators and measures for the goal model, i.e. to set the basis for designing the PI Structure in Step D. Moreover, defining and documenting the indicators and their corresponding measures appropriately is a prerequisite for the quality of the analysis to be performed. For that reason, repeating Activity C1 is a possibility when some questions need to be clarified for achieving an improved definition of indicators and measures (Figure 57). The documentation of the indicators and the measures is in the form of specification tables.

Explanation of Activity C2 based on theory

The specification of indicators and measures is supported by the theory provided in Section 4.5. Several models and theories have been described which cover the essential attributes of indicators and measures, their relationship as well as ways for representing them in line with the goal model. The focus of Step C and especially Activity C2 is to specify the *indicators' attributes*, determine the *source of information* and establish their *interpretation* when the analysis will take place. Step D, on the other hand, covers the design of the indicator structure.

Except from the theory regarding the properties of KPIs, the ISO 15939 Measurement Information Model, analyzed in Section 4.2 (Figure 31, repeated here in the Figure next to this paragraph), provides a graphical guideline of what is to be determined in this activity and a hierarchical flow between the main notions. In Activity C1, the questions referred to the *attributes* of the entities to be measured. The attribute, as it is observed in the ISO 15939, comprises the start for identifying the measures leading to the indicators. As it has been already stated, the term 'indicator' in the EAGQ method is adopted by SMO (Table 23), which implies that, while the GQ(I)M approach is applied, the GQ(I)M's definition of indicator is not accepted.



In both GQM and GQ(I)M methodologies, the reply of the questions is expected to be a measure (or metric). In the ISO 15939, the attributes are related to base measures only. Despite these facts, in the EAGQ method, it is expected that the answer to a question can be a more complex measure which will then be decomposed to simpler component measures.

As it was explained in the beginning of the chapter (Section 5.1.1.), the term 'measure' refers to both simple and more complex measures. Thus, in the EAGQ method, the questions can be answered with a measure in general.

Based on the GQ(I)M approach, the indicators are viewed as an intermediary concept for better identification of measures. In the EAGQ method the indicators still have the power of driving the selection of measures but for a different reason. While answering the questions, the ability of the measures to determine the indicator of the corresponding goal has to be considered. In other words, for determining the indicator, the feasibility of deriving the measures from the EA model has to be considered. By viewing the measures as parts of the indicator that expresses a goal and determining them accordingly, performing unnecessary measurements can be avoided.

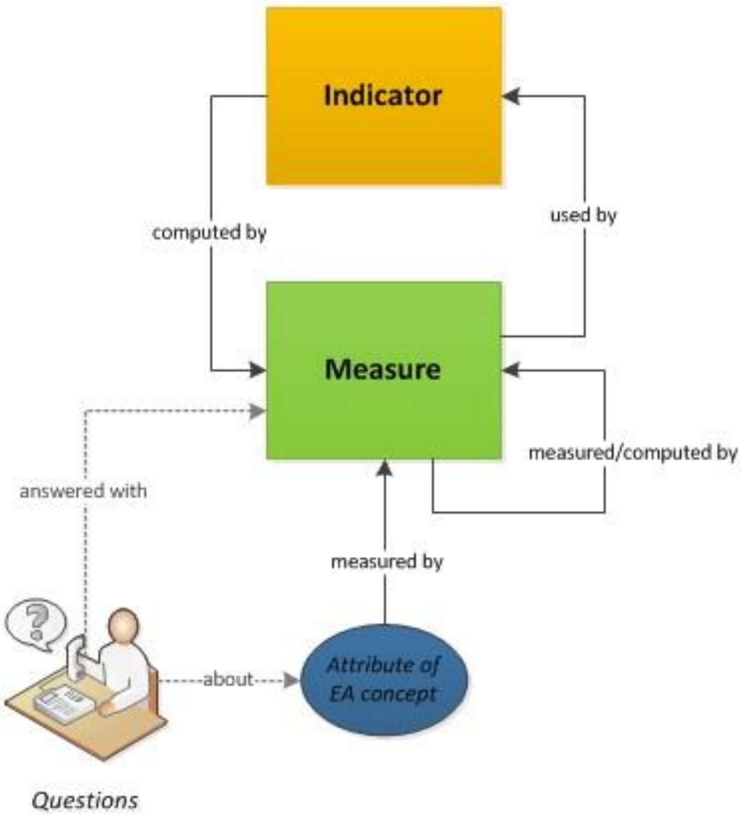


Figure 58 - Questions, Measures, Indicators and Attributes in EAGQ method

In the EAGQ method, the measures have a twofold purpose. On the one hand, measures comprise the answer to the questions regarding the attributes of an EA concept. On the other hand, they are used for computing either an indicator's value or another measure's value.

An overview of the explanation is provided in Figure 58 (see also Figure 48). Figure 58 demonstrates the view of the EAGQ method regarding the relations between attributes of the EA concepts, measures and indicators as well as the role of questions. This hierarchy is also aligned with Figure 49 and comprises a predecessor of the PI Structure presented in Step D.

Description of the tasks and outputs of Activity C2

The tasks that comprise Activity C2, i.e. the indicator and measure assignment process, are:

a) *For the level-value 0 goals:*
   - Determine *measures* that answer the questions per goal
   - Identify *indicators* based on the measures that best express and facilitate the analysis of the measures
   - Check the alignment between the indicator and the corresponding goal

b) *For higher-level goals:*
   - Identify *indicators* based on the indicators determined for the lower level goals
   - Check the alignment between the indicator and the corresponding goal

c) *For each indicator:*
   - Determine each indicator's attributes.

The ***Indicator Specification Table*** (Table 42, end of the Activity C2 description) contains all the attributes that need to be defined for each indicator and provides explanation for each one of them. The quantification-specific and analysis-specific elements are also indicated.

Quantification-specific attributes:

- Scale
- Measures
- Calculation formula

Analysis-specific attributes:

- Decision criteria:
  - Target value & Planned values (if relevant)
  - Threshold & tolerance values
  - Worst value
  - Timeframe
- Performance levels – Interpretation

An example of the Indicator Specification Table is given in Chapter 6 (Table 69). The template of the *Indicator Specification Table* is provided in APPENDIX VI.

- Determine each measure's attributes.

  The ***Measure Specification Table*** (Table 43, end of the Activity C2 description) includes all the attributes (dimensions) that need to be defined regarding a measure including the measurement approach that will be performed. In Section 4.5.2 (Figure 33) four dimensions were identified for determining a measure. Since, there is no need for specifying between a derived, base or aggregated measure, the dimension 'Number of instances' is dropped. This dimension is replaced by the identification of the component measures in case the measure is derived from or computed by other measures. In such a case, the component measures have to be stated in the table as well as their corresponding measurement approaches. The component determination process is not needed when the measure to be specified is a simple measure.

  An example of the Measure Specification Table is given in Chapter 6 (Table 70). The template of the *Measure Specification Table* is provided in APPENDIX VI.

- Take into consideration the 'Goal specification Table' (output of Activity A2).

  The *target value* of the goal should be aligned with the target value of the indicator. In addition, the evaluation type of the goal should also be aligned with the indicator's formula output.

- Check the alignment between the indicators of level n and n-1.

  The main assumption of this model is that the indicators defined in level n-1 can be used for defining and calculating the indicators in level n. This assumption encourages reusability, reduces complexity along the measurement and analysis path and promotes a more practical decomposition of goals.

Finally, to facilitate the building of the PI Structure in Step D, for every indicator defined per goal, a pair could be noted in a table such as Table 40.

Goal – Indicator Pairs	
Goal	**Indicator**
Achieve lower price	Achieved price reduction per item
Improve delivery time	Delivery time reduction

Table 40 - Goal-Indicator Pairs table

Activity C2. Assign indicators to each goal	
Goal	To specify indicators' and measures' attributes, determine the source of information, establish their interpretation and assign them to the corresponding goals
Stakeholders	*IT Management*
Assumptions	• The indicators defined in level n can be used for calculating the indicators in level n+1. • The measures that are needed for calculating the indicators can be derived from the EA model.
Input	**Output**
• List of questions for every goal with level-value equal to 0 • Target Goal model • Goal Specification Table	• Indicator Specification tables • Measure Specification tables • Goal-Indicator pairs
Techniques	
• GQM • GQ(I)M: Steps 7-8 • KPI Properties & Visualization • ISO 15939 Measurement Information Model • Benefit Specification Matrix (Figure 25) • BIM Approach: Conversion Factors & Range Normalization • Financial Analysis Techniques and Performance & Availability measures Overviews (Figure 23 and Figure 24)	

Table 41 - Overview of Activity C2

<table>
<tr><th colspan="3">Indicator Specification Table</th></tr>
<tr><td></td><td>Benefit</td><td>Benefit Specification Matrix
<table>
<tr><th>Classification change</th><th>Measurement of effect</th><th>Probability</th></tr>
<tr><td>• Do new things
• Do things better
• Stop doing things</td><td>• Financial
• Quantifiable
• Measurable</td><td></td></tr>
</table>
Explanation: This property identifies the indicator as a ‘benefit’. The Benefit Specification Matrix (Figure 13) can be used for defining the benefit. From the benefit characteristics included in this matrix, the ‘time span’ and ‘frequency’ coincide with the indicator properties ‘timeframe’ and ‘frequency of measurement’, respectively. It is worth to emphasize that the ‘Measurement of effect’ of the indicator is going to be the same for every architecture alternative (in order to be comparable), but the ‘Classification change’ may be different due to different approaches on the solution.

In case the indicator’s purpose cannot be considered a benefit, this property can be left blank.</td></tr>
<tr><td rowspan="3">Quantification-specific attributes</td><td>Scale</td><td>Explanation: The scale refers to the measurement unit.</td></tr>
<tr><td>Measures</td><td>Explanation: This table contains only the names of the measures that are going to be used for calculating the indicator. The specification of the measures is presented in separate tables. In case, the level-value of the goal is higher than 0, then this property should mention the names of the lower-level indicators.</td></tr>
<tr><td>Calculation formula</td><td>Explanation: The calculation formula of the indicator represents the analysis model as it is defined in Table 23. It is the mathematical expression used for quantifying the indicator.
a) For the level-value 0 goals: The formula uses measures as input.
b) For higher-level goals: The formula uses lower-level indicators as input.
The use of conversion factors may be useful for combining the values of the component measures or indicators of the formula (BIM Conversion Factors method).</td></tr>
<tr><td rowspan="3">Analysis-specific attributes</td><td>Target value → pl scale = 1</td><td>Explanation: The target value of the indicator inherits the target value of the goal. It is one of the decision criteria that are going to be used during the analysis. The target value corresponds to value ‘1’ of the performance level scale as it was introduced by the BIM Approach in Figure 42 and the BIM Range Normalization method.</td></tr>
<tr><td>Planned values</td><td>Explanation: This property is not necessary to be defined. The planned values refer to intermediate values, determined by the stakeholders, which should be achieved during the migration to the solution. They are used as monitoring points for the progress of the project.</td></tr>
<tr><td>Threshold value → pl scale = 0</td><td>Explanation: The threshold value is also part of the decision criteria and represents the value ‘0’ in the performance level scale.</td></tr>
</table>

Analysis-specific attributes	**Worst value** → pl scale = -1	*Explanation:* The worst value represents the lowest value that the indicator could result in. The worst value is one of the decision criteria and represents the value '-1', i.e. the lowest boundary of the normalization formula, in the performance level scale.
Analysis-specific attributes	**Tolerance value**	*Explanation:* The tolerance value does not play a direct role in the performance level scale, but it belongs to the decision criteria. It indicates the flexibility of assigning an indicator value in a level region.
Analysis-specific attributes	**Performance level – Interpretation** *Explanation:* After the performance level values have been determined, two performance regions have been determined: [-1, 0] and [0, 1]. The performance regions can be further refined for expressing a more detailed and accurate assessment of the indicator value. Next to splitting the regions, an interpretation of each one of them is also necessary.	<u>*Example*</u> [-1, 0] : not worthy [0, 0.2] : basic/minimum [0.2, 0.5] : progressing [0.5, 0.8] : very good [0.8, 1] : best
Analysis-specific attributes	**Timeframe** *Explanation:* While the target value of the indicator is the same with the corresponding goal's target value; the indicator's timeframe depends on the timeframe of the target EA alternative that is analyzed. Thus, the timeframe needed for an indicator to be achieved is determined by the time span of the investment option being explored, i.e. the time until the proposed solution is functional.	Since time for achieving an indicator is an important factor for evaluating the respective EA alternative, the difference between the timeframe of the goal and the indicator can be used as a *multiplier* for determining the *final performance level* of the indicator: $$time\_difference\ (td) = \frac{t_i - t_g}{t_g}$$ Where: t_i is the timeframe of the indicator t_g is the timeframe of the goal Then, the *final performance level* is: $$final\ pl = pl - pl * td$$ so, if $td > 0 \Rightarrow final\ pl < pl$ if $td < 0 \Rightarrow final\ pl > pl$
	Frequency of measurement	*Explanation:* The frequency is an attribute referring to how often the performance of the indicator should be calculated. It can be defined according to the timeframe as well as the 'planned values' property.
	EA layer	*Explanation:* The 'EA layer' attribute refers to the architecture layer addresses by the indicators. The determination of the layer(s) is based on the monitoring objects of its measure(s).
	Visualization	*Explanation:* The last property of the indicator refers to its visualization. As it is explained in Activity C3, there is set of available visualizations as well as guidelines for selecting the most appropriate one depending on various factors. Priyanto's Master Thesis [115] provides more detailed information on the topic.

Table 42 - Indicator Specification Table: Explanation

Measure Specification Table	
Measure's name or ID	
Type of measure	*Explanation:* The type of measure refers to the measurement approach followed for obtaining the value of the measure. The outcome of the measurement approach can also be determined, e.g. time, count, condition, etc., which actually denotes the scale of the measurement. For determining the measurement approaches of the measures, the overviews of the financial and performance EA analysis techniques provided in Chapter 3 are helpful.
Component measures	*Explanation:* In this section, the names of the component measures are mentioned. The component measures can be identified by considering: • techniques such as the TCO approach or the ABC method in case of financial analysis. Table 75 can also be used as guideline for identifying cost elements which can serve as component measures. • performance and availability measures. By knowing which of them have been already used in EA analysis and which others are suitable for EA analysis, the component measures can be determined.
Type of scope	*Explanation:* For the measure specified in this table as well as its component measures, the type of scope should be determined too. According to Cardoso [103], the type of scope can be: • *temporal scope* (time-based) • *number of instances* selected for the calculations • *process state scope* that selects the instances that are in a given process state
Type of monitoring objects and points	*Explanation:* The measures refer to specific entities in the architecture model. Having built the EA analysis model in parallel with Activity C1, as suggested, is beneficial for mapping the measures to the objects to be measured and the points where the data are gathered. A view of the specific architecture model indicating which the contributing concepts are would be more clear and precise instead of listing all the participating concepts. Moreover, being able to indicate which are the concepts that the measures are related to, provides proof that the data needed for calculating the indicators can be derived from the architecture model (Assumptions, Table 41).

Table 43 - Measure Specification Table: Explanation

Activity C3. Define visualizations for the indicators

Since the analysis is also addressed to Boardroom stakeholders, except from the IT management, visualizations are important for getting the essence of the analysis outcome and for providing a meaningful overview of the progress achieved through the transformation. Additionally, visualizations can assist in comparing different alternatives with one another and realizing the difference between the baseline architecture and a target architecture.

Priyanto [115] in his Master thesis describes the selection of visualization alternatives and representations of metrics and KPIs as part of his EA-based Decision Making Method. He proposes a formal documentation of a metric in his framework which includes, among other information categories, information about the visualization of the metric. The *visualization type* is selected based on the classification of the possible visual outputs, which depend on the kind of 'message' to be conveyed as the result of the analysis. The possible types are: comparison, distribution, composition and relationship (APPENDIX VII). Moreover, for every visualization type, there are *visualization options* for determining the most suitable chart type, such as line charts, bar charts, pie charts, etc.

Thus, in this activity, based on Priyanto's framework, the most appropriate visualization type and option of the indicators is determined. The visualization types should be accompanied by an explanation of the purpose of the visualizations. The input for performing this activity includes the indicator and measure specification tables where the scale of the indicators and the type of value output expected from the measurements are determined. Table 44 summarizes Activity C3.

Activity C3. Define visualizations for the indicators	
Goal	To select the most suitable visualization type(s) per indicator for demonstrating the analysis outcome
Stakeholders	*IT Management*
Assumption	The indicator has been defined properly and its expected values are also clear in terms of type and number of instances.
Input	**Output**
• Indicator Specification tables • Measure Specification tables	• Visualization types per indicator
Techniques	
• Priyanto's EA-based Decision Making Method	

Table 44 - Overview of Activity C3

Activity C4. Determine the Focus area of each goal in the Baseline architecture

Based on the assignment of monitoring objects to the measures of the indicators, the definition of the *focus area* of each goal in the goal model becomes a trivial task. Defining the EA domain and the specific concepts that the goal refers to indicates the influence that this goal has on the architecture model as well as which EA concepts contribute to the realization of the goal. The essence of this activity is to follow the traceability links between a goal and the EA concepts which, eventually, realize it.

To assign EA elements to goals, the user has to:

a) Isolate the path from the goal to the requirements that realize it
b) Identify the concepts that realize the requirements through the measures assigned to them
c) Roll back the target goal model and assign the corresponding EA concepts to the goal

Since the architecture model is different for the baseline architecture and for target architectures, the focus area of the goal also changes. Therefore, this activity is repeated in Step G. An example of the focus area model of a goal is depicted in Figure 59 for the goal ‘Improve delivery time’.

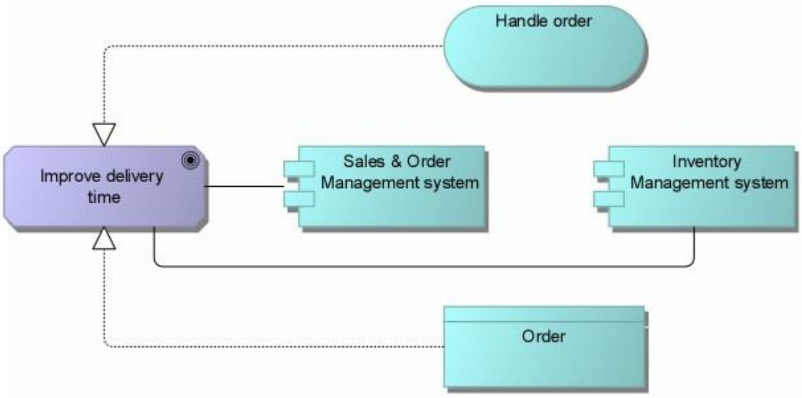
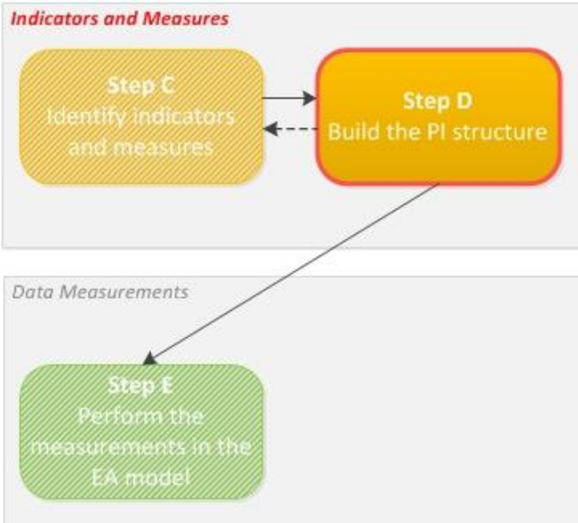


Figure 59 - Focus area example: Improve delivery time

Activity C4. Determine the Focus area of each goal in the Baseline architecture	
Goal	To identify the part of the EA model that is influenced by each goal
Stakeholders	*IT Management*
Input	**Output**
• Target Goal Model • EA concepts - Rqs relationships • Indicator Specification tables • Measure Specification tables	• Focus area per goal
Techniques	
• Goal Properties Identification	

Table 45 - Overview of Activity C4

Step D: Build the PI structure



After defining the indicators and their corresponding measures, the next step is to create the PI structure. The PI structure is a model of indicators derived from the goal model. The relations among indicators have to reflect the relations among goals and the relations between goals and requirements. To provide a more complete representation of the quantification model, apart from the indicators, the measures that are calculated in the EA level are also depicted. The artifacts of Step C are of high importance for developing the main product of this step, i.e. the PI structure. Preliminary actions are also discussed, which entail the discovery of redundant or duplicate measures (Figure 60).

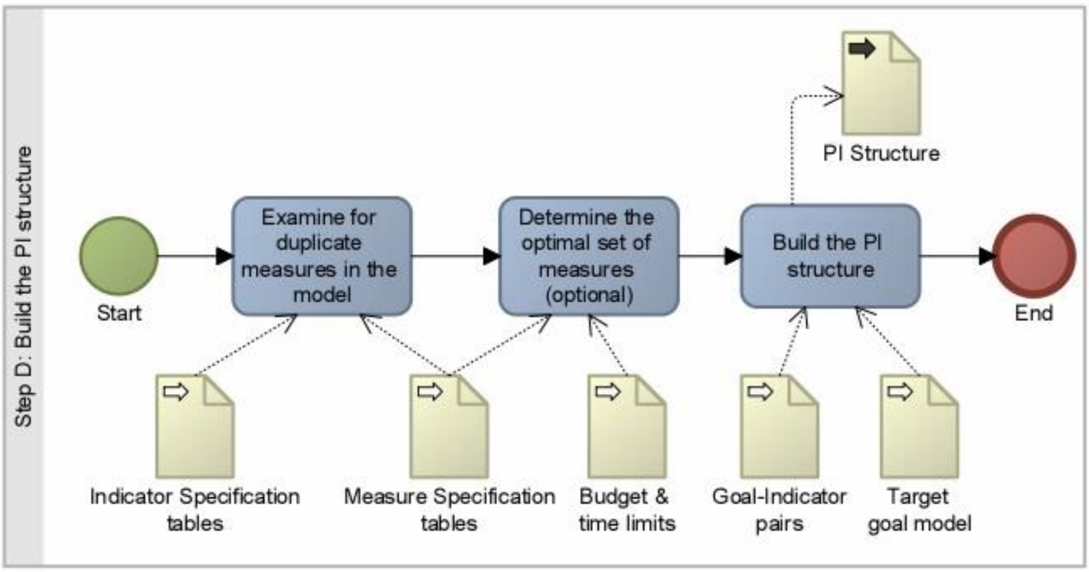


Figure 60 - Step D: Activities, Input & Output data

Activity D1. Examine for duplicate measures in the model

The purpose of this activity is to analyze whether duplicate measures exist. Duplicate measures should be identified and eliminated before building the structure. The user should perform this check while answering the questions and determining the measures assigned to requirements. In case such an action has not taken place, this activity is responsible for avoiding redundant calculations and measurement repetitions.

The specifications of the measures that were determined in the previous step serve as input for this activity. One of the reasons that redundancy may exist is the different naming of measures in large goal models with many requirements. To make sure that two measures are the same, the specification tables of the measures should be examined, including their monitoring objects and points. If the measure dimensions are the same, then one measure should be assigned to all affected indicators (and consequently goals), while the duplicates should be eliminated.

Activity D1. Examine for duplicate measures in the model	
Goal	To examine whether duplicate measures exist and eliminate the redundant ones.
Stakeholders	*IT Management*
Assumption	This activity is performed in case the check for duplicate measures has been neglected during the specification of the measures.
Input	**Output**
• Indicator Specification tables • Measure Specification tables	• List of unique measures • Updated Indicator and Measure Specification tables
Techniques	
No specific technique applies in this activity, except from *comparison* among measure specifications.	

Table 46 - Overview of Activity D1

Activity D2. Determine the optimal set of measures (optional)

This activity concerns the definition of the optimal set of measures. The Optimum Measures Set Decision (OMSD) Model, introduced in Section 4.1.4, provides a methodology for selecting an optimal set of measures based on various factors such as cost, resource and time constraints (Figure 30). This activity can be proven helpful in cases where there is a limited budget or time for making decisions regarding the selection of the target architecture. As it has been mentioned in the description of the TCO approach, the costs for performing the measurements as well as the resources consumed for realizing the measurements, most of the times are neglected. Hence, taking into consideration the factors proposed by the OMSD model can reduce the risk of defining measures that do not comply with the current decision-making process and the availability of resources.

The determination of the optimal set of measures, though, is optional. While not performing this activity will not influence the remaining steps of the EAGQ method, performing it will influence the flow of the method. In case the defined optimal set of measures is smaller than the initial set of measures, then Step C needs to be repeated. The interdependency of Steps C and D is denoted by the arrows between the two steps in Figure 61. Since this activity is optional, the arrow leading back to Step C is dashed.

Indicators should be assigned to all goals for facilitating their quantification, but some of the indicators may be too detailed, demand complex measurements or data hard to find. That is why activities C1-C3 should be revisited and simpler measurements should be defined.

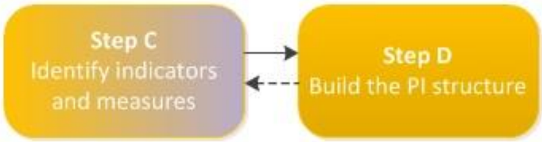


Figure 61 - Interdependency Steps C and D

The tasks comprising Activity D2 are:

a) Application of the OMSD model, i.e. selection of measures based on *cumulative cost* and *effort* needed for collecting the data and performing the measurements
b) After the optimal set selection is completed; ensuring that *all indicators* are *covered* by repeating Step C.

Activity D2. Determine the optimal set of measures (optional)	
Goal	To reduce the set of measures based on cost and effort for performing the measurements.
Stakeholders	*IT Management*
Input	**Output**
• Measure Specification tables • Input values for selection factors: budget, time, resource constraints	• Optimum set of measures
Techniques	
• Optimum Measures Set Decision (OMSD) Model	

Table 47 - Overview of Activity D2

Activity D3. Build the PI structure

The last activity of Step D is also its core activity. The PI structure is a model which represents the computational relationships among the indicators identified in Step C. While the PI structure introduced by Popova and Sharpanskykh [102] includes only the indicators, the PI Structure designed in the EAGQ method contains both indicators and measures. This modification is proposed in order to be able to show the source for the calculations of the indicators computed directly from measures and not from other indicators. Clarifying the two types of indicators:

- Indicators calculated by other indicators: Such indicators are assigned to goals which relate to lower-level goals through aggregation relations.
- Indicators calculated by measures: Such indicators are assigned to goals that are directly realized by requirements.

Since the output of Step C included the pairs of goals and indicators, building the structure is a rather easy task. The relationships, though, between the elements of the PI structure need to be clarified. So, the rules for deriving indicator relationships from goal relationships are explained in Table 48 and illustrated in Figure 62.

EA concepts	Goal model relation	PI structure entities	PI structure relation
Goal-to-Goal	**1. Aggregation**	Indicator-to-Indicator	**1. Aggregation (computation)**
	2. Influence && Computational expressivity		**2. Aggregation (computation)**
	3. Influence && Not Computational expressivity		**3. Association**
Goal-to-Requirement	**4. Realization**	Indicator-to-Measure	**4. Aggregation (computation)**

Table 48 - Relations Derivation Rules: From the goal model to the PI structure

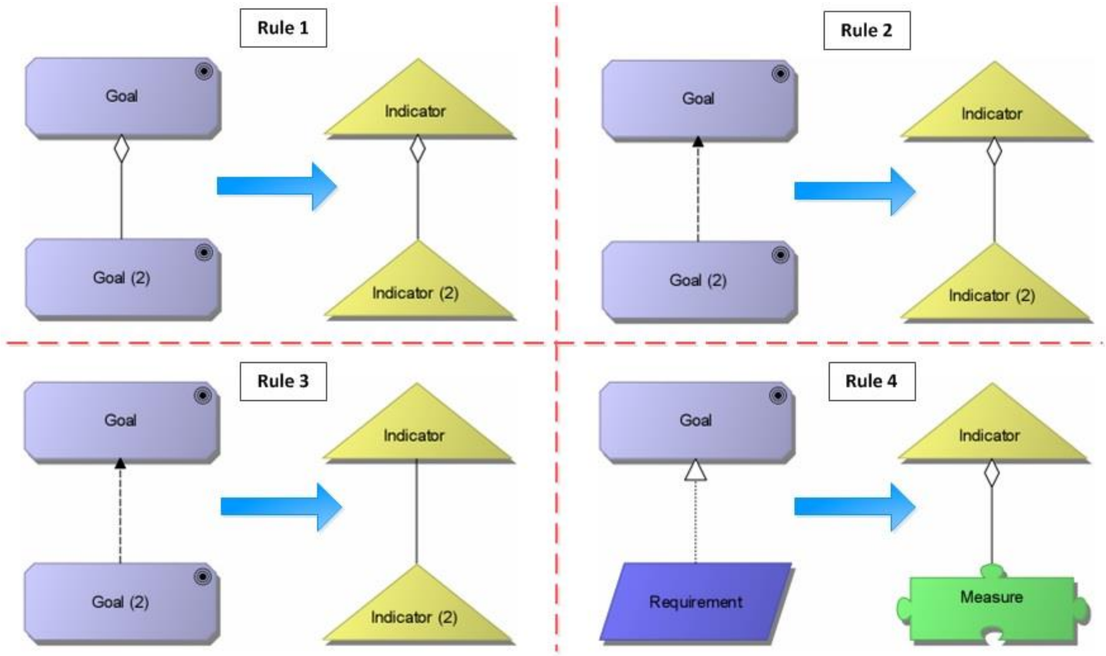


Figure 62 - Relations Derivation Rules: Graphical representation

It can be observed that there are combinations of EA concepts and relations missing from Table 48. These combinations regard the influence relation between requirements and the influence relation between a goal and a requirement. Since there is no certainty whether the influence relation can be translated into a computational relation in terms of indicators, the same rules that apply for the influence relation in Table 48, apply in such situations. Hence, it depends on the computational expressivity of the influence relationships. The inability of computational expressivity between any pair of concepts can be still utilized for influencing the certainty (probability) of the performance of the indicator to which is related. For instance, a high performance level of the influencer indicator can increase the probability of the influenced indicator, while a low performance level could indicate difficulties in gaining the expected benefits from the influenced one.

A template of the PI structure as an output of this model is presented in Figure 63. The graph has been developed in the BiZZdesign Architect tool. In the structure:

- Triangles represent indicators:
  - yellow-colored indicators denote the computability of the indicator with regard to its higher-level and lower-level indicators
  - the blue-colored indicator represents an influencer with no computational expressivity
- Green puzzle-pieces represent the measures in the requirement level
- Orange puzzle-pieces represent the measures in the EA level

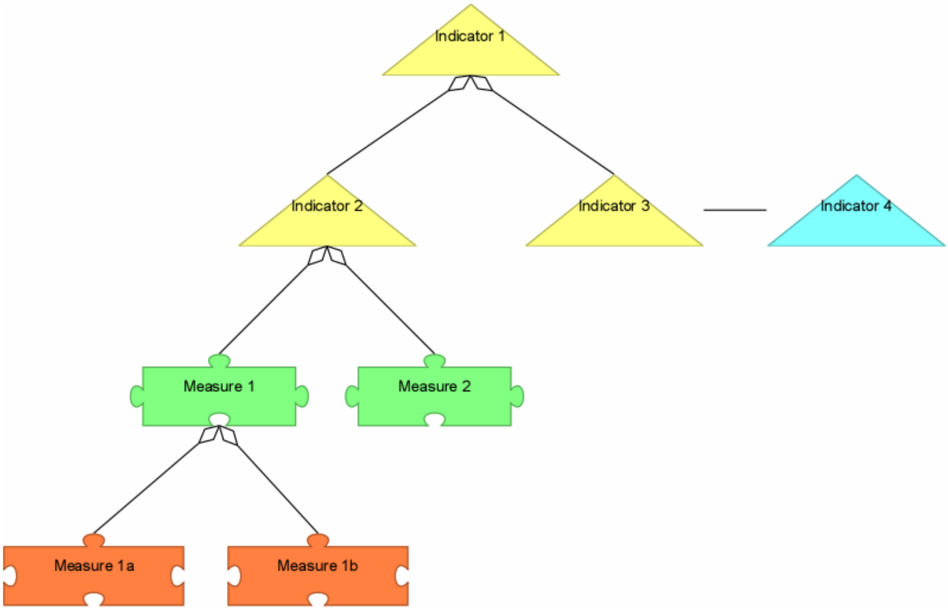


Figure 63 - EAGQ method PI Structure illustration

Now that all three types of models discussed in the EAGQ method have been introduced, how the models relate with each other needs to be clarified. The PI structure comprises the link between the goal model and the EA model in the analysis space and facilitates the quantitative analysis of the goal model by propagating the information derived from the EA model to the goal model through the performance indicators (Figure 64).

Activity D3. Build the PI structure	
Goal	To design the indicator model for representing the computational relations among indicator and measures.
Stakeholders	*IT Management*
Input	**Output**
• Goal-Indicator pairs • Target Goal Model • Indicator Specification tables • Measure Specification tables	• PI Structure including indicator-to-measures relations
Techniques	
• PI Structures • BIM Approach	

Table 49 - Overview of Activity D3

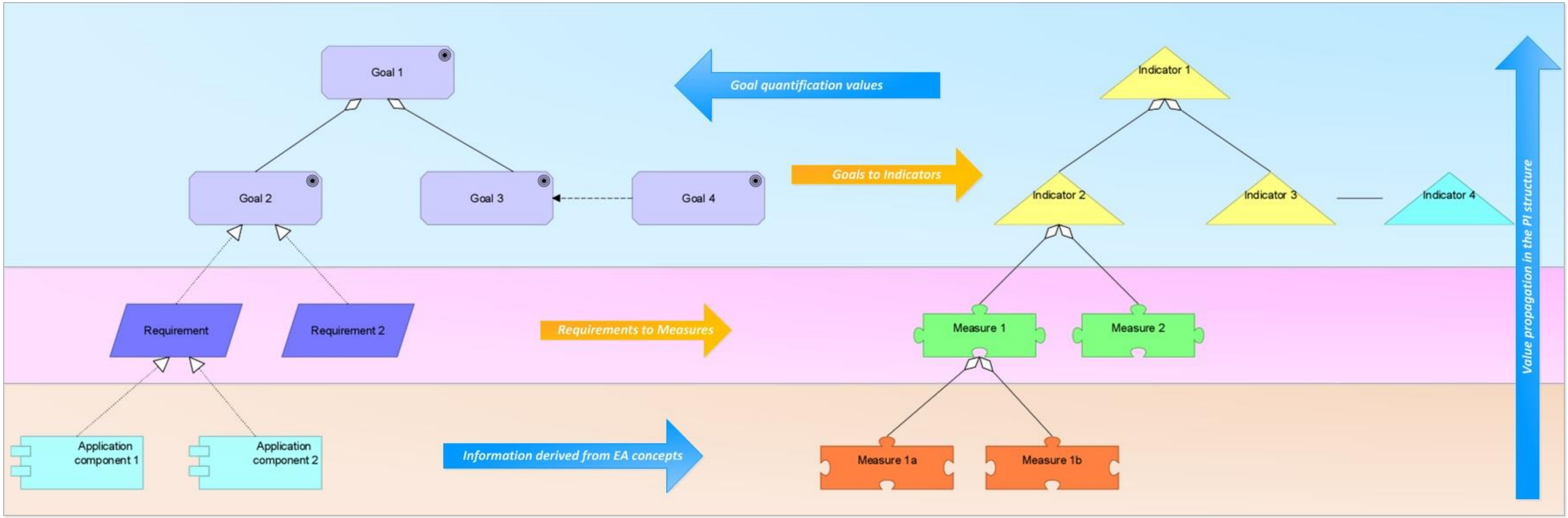
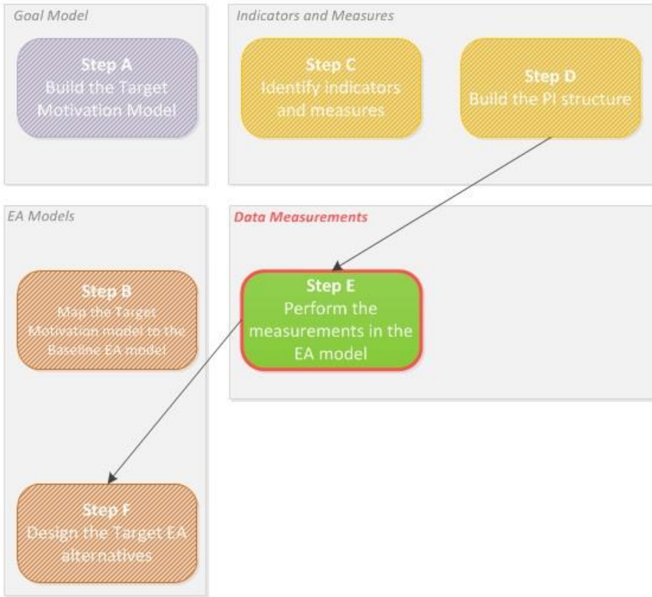


Figure 64 - Relation between the Goal model, EA model and PI structure & Information flow across the models[5]

[5] The blue arrows demonstrate the information flow and the value propagation across the models, while the orange arrows show the correspondence among elements of the Goal model and the PI structure. See also Figure 49.

Step E: Perform the measurements in the EA model



Completing the first four steps of the EAGQ method denotes that all the preparations have finished and the moment for performing the actual measurements has arrived. The measurements take place in the EA model. So, considering the transformations quadrant, there is change of quartile but vertically in the analysis space. The EAGQ method moves from the goal analysis space to the architecture analysis space. In Step E, the main concerns are the design of the EA analysis model(s) as well as the application of the measurement methods and functions (Figure 65). Therefore, the main deliverables are the EA analysis models and the computations of the measures.

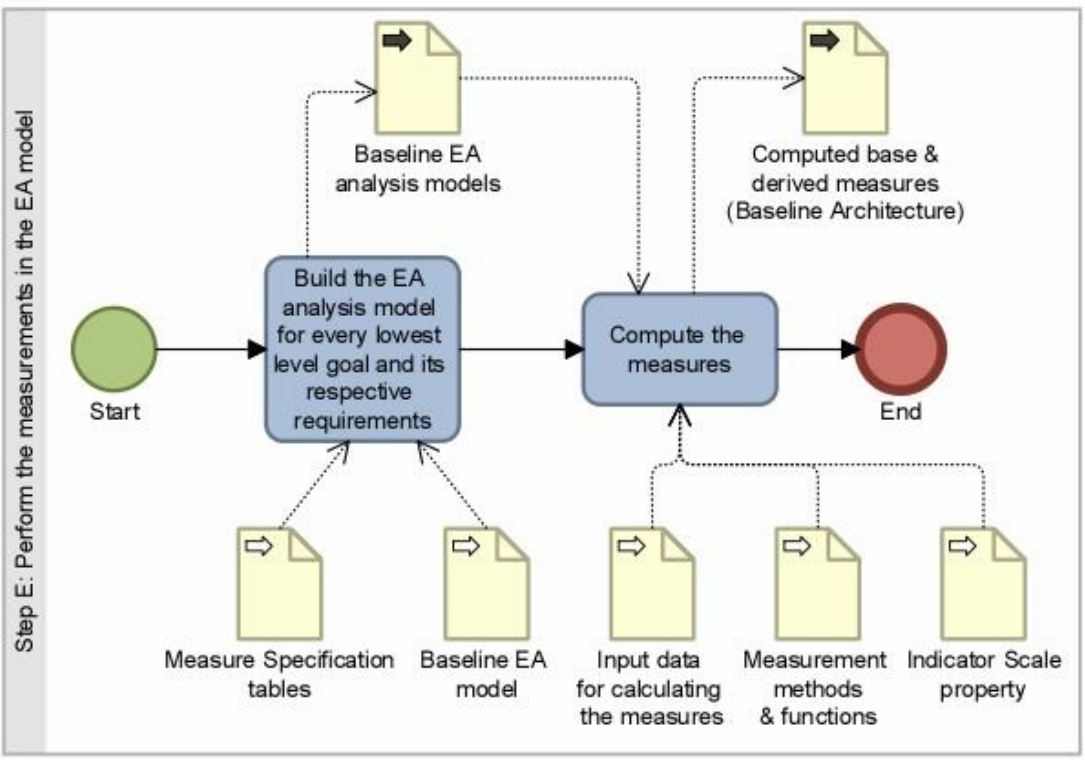


Figure 65 - Step E: Activities, Input & Output data

Activity E1. Build the EA analysis model for every lowest level goal and its respective requirements

The analysis models comprise sub-models derived from the EA model, which include all the EA concepts that participate in the measurements. The decision about which elements an analysis model should consist of depends on the defined measures. The measure specification tables defined in Step C and refined in Step D comprise the input for this activity.

Examples of EA analysis models have been discussed in a series of papers by Jonkers, Iacob and their colleagues (e.g. [28] [38] [116]). These models are parts of the Performance & Cost EA Analysis method, discussed in Section 3.8.1, where they are used for facilitating the calculations of performance measures and costs across the EA model.

As a reminder, the measures are assigned to requirements and consequently to the level-value 0 goals; hence, the analysis models can include these motivational extension concepts to demonstrate the purpose of each analysis model.

The analysis models at this step of the EAGQ method are derived from the Baseline EA model due to the fact that the measurements of the Activity E2 concern the baseline (current) situation.

<table>
<tr><th colspan="3">Activity E1. Build the EA analysis model for every lowest level goal and its respective requirements</th></tr>
<tr><td>Goal</td><td colspan="2">To design the Baseline EA analysis models for supporting and performing the measurements.</td></tr>
<tr><td>Stakeholders</td><td colspan="2">IT Management</td></tr>
<tr><td>Assumption</td><td colspan="2">All useful concepts for the measurements that are going to take place can be found in the Baseline EA model designed in Step B or related views of the model.</td></tr>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td colspan="2">• Measure Specification tables
• Baseline EA model</td><td>• Baseline EA analysis models</td></tr>
<tr><th colspan="3">Techniques</th></tr>
<tr><td colspan="3">• Performance & Cost EA Analysis method</td></tr>
</table>

Table 50 - Overview of Activity E1

Activity E2. Compute the measures

For performing the measurements, the first task in this activity is to assign input data for the specified attributes of the EA concepts. Then the measurement methods and functions for calculating/computing the measures have to be applied.

The measurements as well as the techniques that are going to be used for performing them should be derived from the specification of the measures. As it is described in Step C, while defining the 'type of measure' dimension, the overview of available techniques should be considered. Of course, not all possible techniques are described in this thesis, but the discussed methods provide guidelines for understanding possible ways to perform measurements. Another measure dimension that contributes to the identification of the technique that should be used is the 'type of scope', while the 'scale' of the corresponding indicator that the measure belongs to should be also taken into account.

Depending on the analysis type needed in the indicator level, a related computational process is expected in the measures level. For example, for financial analysis in the indicator level, cost identification and calculation would be suitable in the architecture level. While for availability analysis in the indicator level, performance measures such as failure rates, utilization and workload should be addressed in the architecture.

Techniques that can contribute in cost identification are the TCO as well as the Life Cycle Costing approach which have been combined in APPENDIX II, Table 75. The EA layer where the components under analysis belong to helps in focusing on the important costs for the specific layer. Additionally, techniques regarding cost calculations are, for example, the ABC approach, the Time-driven ABC, Software cost estimation approaches or even Time Savings Time Salary model. The latter combines two analysis approaches, since it is based on time which is then converted into cost savings.

While estimating the input values of various concepts is possible with the techniques mentioned above, the propagation of the calculated values across the architecture model for providing the output needed in the requirements' level, i.e. the measures' level, is essential. Such techniques comprise the 'Performance & Cost EA Analysis' approach and the 'Cost allocation across EA models' approach (Chapter 3).

It is also possible that input values can be directly assigned to measures, since the measurement methods for a measure may not be able to be applied in the architecture model. For example, a value for a measure may be the result of a survey. The analysis of the survey data cannot be performed in the EA model, but its outcome can be used for calculating other measures in the requirement-level. Additionally, relating the survey and its outcome with a concept in the EA model provides insight on the reasons for deploying the survey.

When this activity is completed, the values of the measures that correspond to

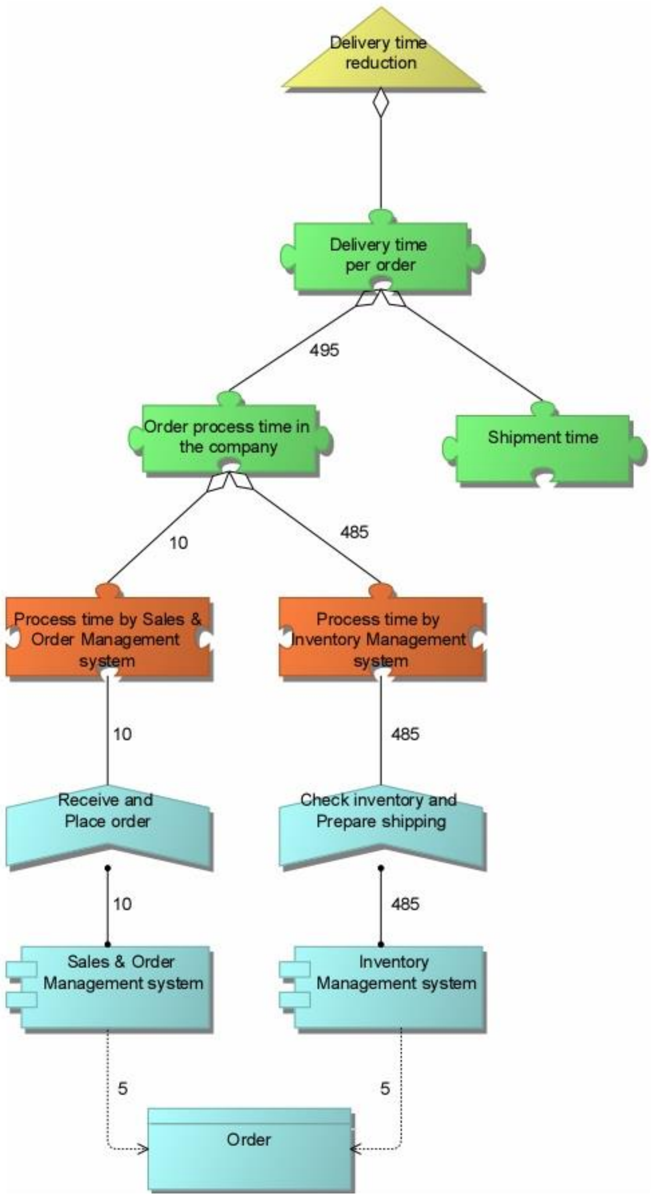


Figure 66 - Step E example: EA analysis model and measurement propagation

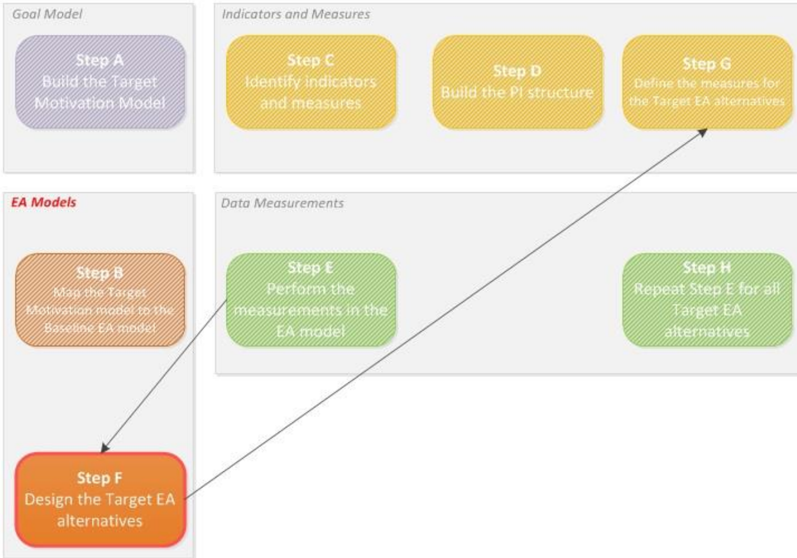
the requirement level for the baseline architecture will be available and will be stored for usage in the indicator analysis in Step I.

An example of the output of Step E is illustrated in Figure 66. The figure presents the EA analysis model for the 'Improve delivery time' goal along with a part of the PI structure. Only the branch regarding the process time has been designed. The measurements in the EA model refer to process time and access time to the data. For the inventory management system, the value '480 minutes' indicates the batch processing of the orders. The process times of the two systems are then aggregated as the total process time in the requirement-level (495 minutes).

<table>
<tr><th colspan="2">Activity E2. Compute the measures</th></tr>
<tr><td>Goal</td><td>To calculate the measures through applying measurement approaches for calculating the EA concepts attributes.</td></tr>
<tr><td>Stakeholders</td><td>IT Management</td></tr>
<tr><td>Assumption</td><td>The input data in the EA model are accurate.</td></tr>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>Baseline EA analysis models
Input data for calculating the measures
Measurement methods and functions (Measure Specification tables)
'Scale' from Indicator Specification tables</td><td>Computed measures for the baseline architecture</td></tr>
<tr><th colspan="2">Techniques</th></tr>
<tr><td colspan="2">Financial analysis techniques:
Analysis frameworks
Software cost approaches
Time-based Cost estimation
Performance and Reliability measures
Value propagation and calculation techniques:
EA-specific Analysis Techniques
Performance & Cost EA Analysis method
Cost allocation across EA models</td></tr>
</table>

Table 51 - Overview of Activity E2

Step F: Design the Target EA alternatives



First, for recapping what has been covered so far by applying the first five steps of the EAGQ method, the following statements provide an overview of what has been achieved:

1. the *Goal model*, the *Baseline EA model* and the *PI structure* have been designed
2. an examination has taken place regarding the degree that the Baseline EA model is able to address the target goals of the organization
3. *indicators and measures* for quantifying and analyzing the performance of the goals have been determined
4. *measurements* have been performed for deriving information from the Baseline EA model in order to assess the architecture towards the goals.

While assessing the relation between the Baseline EA model and the goal model, a set of inefficiencies of the current architecture have been discovered. Thus, a target EA model has to be designed in order to address the target goal model and provide solution to the identified problems or gaps. The target EA model will be used for evaluating the improvements of the new architecture compared to the current architecture. It is possible that multiple target architecture models may be able to address the goals by following different approaches. In that case, comparisons among alternative EA models will also take place. Consequently, the core outputs of Step F are the target EA models as well as the mapping of their core concepts to the requirements of the goal model.

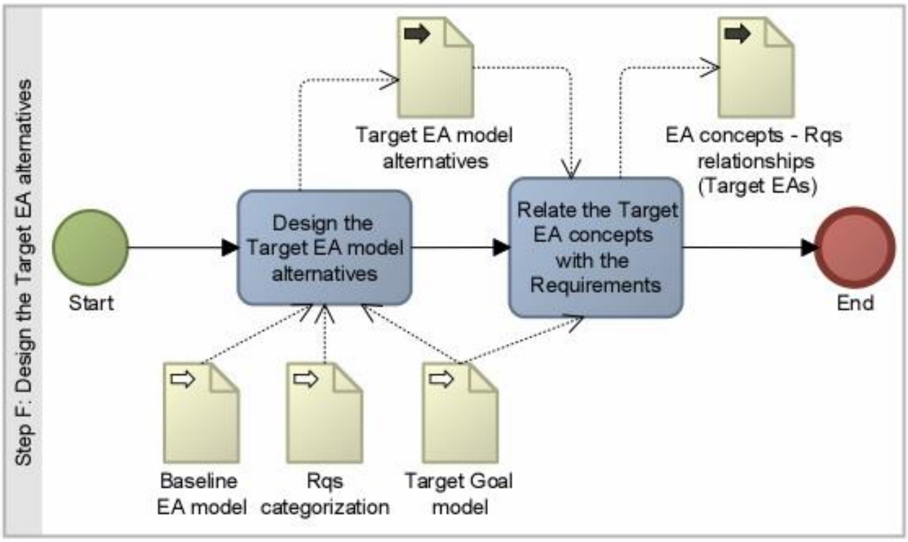


Figure 67 - Step F: Activities, Input & Output data

Activity F1. Design the Target EA model alternatives

The Target EA alternative model scenarios are designed and identified based on the problems or gaps which have been determined in Activity B3 ('Identify problems/gaps in the realization of the Requirements'). The output of that activity, i.e. the categorization of requirements as 'covered', 'partly covered' or 'not covered', is the starting point for deciding which are the requirements that need to be addressed for fulfilling the goals.

The requirements express a need that should be realized by some concepts of the architecture, but it is possible that more than one solutions exist for realizing these requirements. Thus, a set of alternative target EA models could be designed for addressing the requirements. The satisfiability of requirements may vary across the EA alternative models. For example, one alternative may focus on the requirements that are not covered, while another may try to address all three types of requirements. The Baseline EA model should also be available, since re-usability of already existing systems is always beneficial, if possible.

Designing the target EA alternatives requires the involvement of various stakeholders, mainly from the IT management. The stakeholders will also be responsible for estimating the values for performing the measurements in each alternative in Step H.

<table>
<tr><th colspan="2">Activity F1. Design the Target EA model alternatives</th></tr>
<tr><td>Goal</td><td>To build the Target EA model alternatives for facilitating the comparison among them in the indicator level.</td></tr>
<tr><td>Stakeholders</td><td>IT Management</td></tr>
<tr><td>Assumptions</td><td>• The Target EA model alternatives can be designed.
• Proper research has been conducted for recommending solutions that address the target requirements.</td></tr>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>• Target Goal model
• Requirements Categorization
• Baseline EA model</td><td>• Target EA model alternatives</td></tr>
<tr><th colspan="2">Techniques</th></tr>
<tr><td colspan="2">• Interviews with stakeholders and specialists
• Discussions among specialists</td></tr>
</table>

Table 52 - Overview of Activity F1

Activity F2. Relate the Target EA concepts with the Requirements

After the Target EA model alternatives have been built, the relation of the new concepts with the requirements should be demonstrated. Connecting the EA concepts with the requirements provides twofold benefits. Firstly, it indicates how the target architecture contributes to realizing the requirements that are not fulfilled by the baseline architecture. Secondly, as it is explained for the baseline architecture, it offers support for the quantifications in the goal (indicator) level.

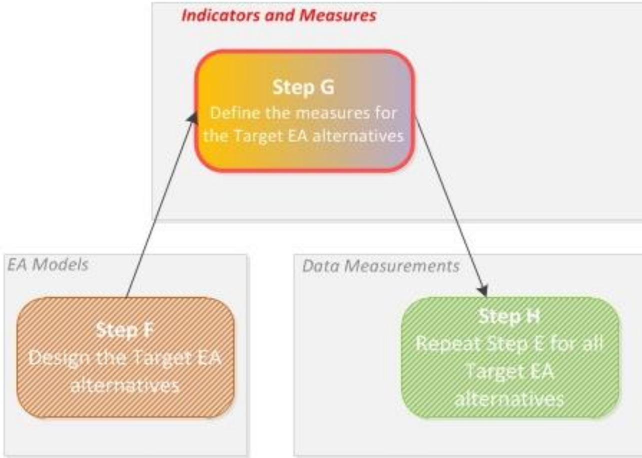
While in Activity B2, two types of relationships were expected to be used between requirements and EA concepts; in this activity the requirements should be related with the EA elements through *realization* relationships. Realization relations denote that the elements address the needs imposed by the requirements. If *association* relations need to be used, this means that still some requirements may not be fulfilled completely. The analysis in the future steps, though, will prove whether it is expected from the target EA alternative to perform better that the baseline architecture.

The stakeholders as well as the assumptions in this activity are similar with the ones in Activity B2 (Table 53).

Activity F2. Relate the Target EA concepts with the Requirements	
Goal	To identify and connect the Target EA concepts for each alternative solution which realize or are related to the target requirements.
Stakeholders	*IT Management*
Assumptions	• The views and the concepts included in the Target EA model alternatives are sufficient for relating the requirements with the concepts • There is the possibility to design other views (more or less detailed) for identifying potential existing realization relationships
Input	**Output**
• Target Goal model • Target EA model alternatives	• Realization and association relationships among Requirements and EA concepts
Techniques	
• Interviews with stakeholders and specialists • Discussions among specialists • Cross-checking of or reference to documentation of requirements, systems, processes, services, databases and data objects, etc.	

Table 53 - Overview of Activity F2

Step G: Define the measures for the Target EA alternatives



While the indicators in the goal domain remain the same, the measure specifications have to be modified due to changes in the architecture domain. The measures are calculated in the EA level and they are dependent on the EA concepts where the data reside.

The value assigned to a measure should follow the specification provided in Step C. Since the sources of the data change, i.e. the monitoring objects, this is the only difference in the specifications of the measures. There is a goal attribute, though, that is also affected by the architecture change, i.e. the focus area of the goal. The focus area denotes the part of the architecture model that the goal influences and, since there are modifications in the architecture, different or new concepts should be assigned to goals. These two observations are the reasons for the two activities of this step.

As an overall comment, Step G serves as an intermediate between the indicator and measure definitions and the measurements to be performed in the target architectures. It aims to adjust the measure specifications for applying them in the target EA models.

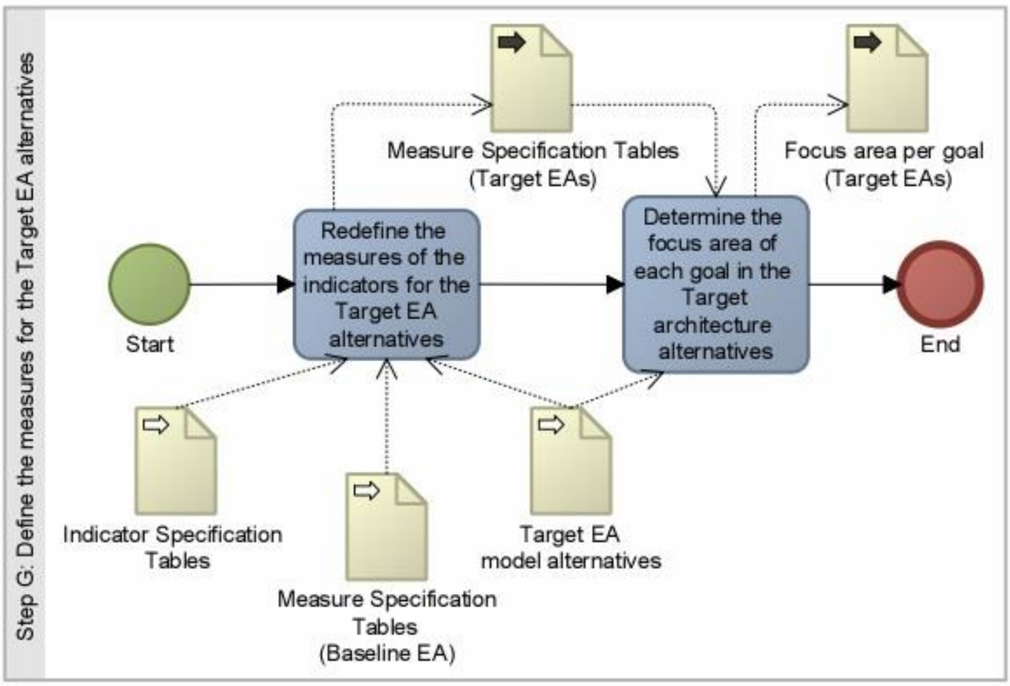


Figure 68 - Step G: Activities, Input & Output data

Activity G1. Redefine the measures of the indicators for the Target EA alternatives

The specification of the measures includes four dimensions, as it is explained in the 'Measure Specification Table' (Table 43). The dimension that is mainly affected by the architectural change is the 'type of monitoring objects and monitoring points' (Figure 69). As the architecture changes, the elements, that need to be measured and analyzed, change as well. Thus, assigning new or different concepts to the measures is essential for providing the right input for the calculations. In line with the proposal in Step C, this activity is facilitated by building the EA analysis models of the target architectures in parallel, which is the responsibility of Step H for the target architecture alternatives. In case more than one target architectures exist, the activity has to be repeated for each one of them.

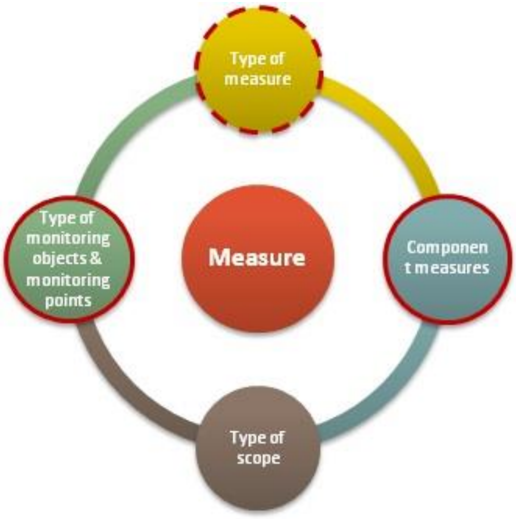


Figure 69 - Measure dimensions affected by Target EA models

Furthermore, the red line around the 'component measures' dimension in Figure 69 indicates that this dimension is also influenced. The change of the monitoring objects may also impose a change on the component measures that are going to be used for the calculations. A change in the component measures respectively causes a change on the 'type of measure' dimension (denoted by the dashed line around this dimension in Figure 69). The effect on the measurement approach of the measures is not mandatory. Most probably the same measurement approaches are going to be used for estimating the measures in the target architectures, but modifications may appear due to particularities of the new elements.

<table>
<tr><th colspan="3">Activity G1. Redefine the measures of the indicators for the Target EA alternatives</th></tr>
<tr><td>Goal</td><td colspan="2">To adapt the measure specifications to the target EA model alternatives.</td></tr>
<tr><td>Stakeholders</td><td colspan="2">IT Management</td></tr>
<tr><td>Assumptions</td><td colspan="2">• The properties of the modified or added concepts are known.</td></tr>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td colspan="2">• Target EA model alternatives
• Measure Specification Tables (Baseline EA)
• Indicator Specification Tables</td><td>• Measure Specification Tables (Target EAs)</td></tr>
<tr><th colspan="3">Techniques</th></tr>
<tr><td colspan="3">• Financial Analysis Techniques Overview (Figure 23)
• Performance and Reliability Measures Overview (Figure 24)</td></tr>
</table>

Table 54 - Overview of Activity G1

Activity G2. Determine the focus area of each goal in the Target architecture alternatives

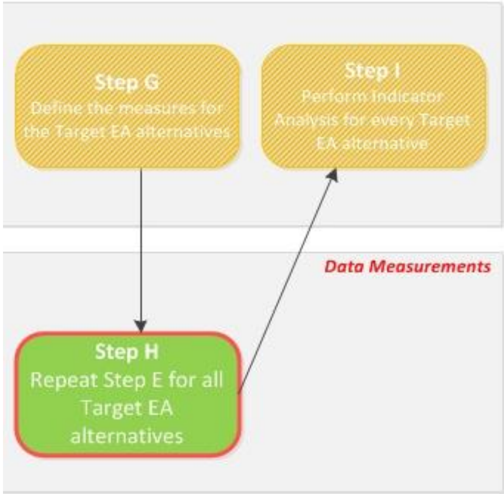
Based on the new assignment of monitoring objects to the measures of the indicators, the *focus area* of each goal can be also redefined. The tasks performed in this activity are the same as in Activity C4. The expected outcome is a set of models (or other types of documentation) that will include the EA concepts which contribute or influence each goal by following the traceability links across the models.

<table>
<tr><th colspan="3">Activity G2. Determine the focus area of each goal in the Target architecture alternatives</th></tr>
<tr><td>Goal</td><td colspan="2">To change the focus area of the goals according to the new measure specifications.</td></tr>
<tr><td>Stakeholders</td><td colspan="2">IT Management</td></tr>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td colspan="2">• Target Goal Model
• Measure Specification tables (Target EAs)</td><td>• Focus area per goal (Target EAs)</td></tr>
<tr><th colspan="3">Techniques</th></tr>
<tr><td colspan="3">• Goal Properties Identification</td></tr>
</table>

Table 55 - Overview of Activity G2

Step H: Repeat Step E for all Target EA alternatives

After the adaptation of the measures specifications according to the target EA models, it is the turn of the target EA models to be measured. The measurement approaches will be applied based on estimated input data. For each proposed solution a set of improvements are expected. These improvements, though, are focused on a specific element of the architecture. By performing the measurement and the analysis, the overall influence of the change will be assessed.



Step H comprises a duplicate of Step E. The only difference is that the input concerns the target architectures instead of the baseline as well as the modified measure specification tables (Table 56).

To continue the example from Activity E2, in the target EA model, the two management systems have been replaced by an integrated one, called ‘Order Management system’ (Figure 70). In the new system, the orders are not processed in batches every 8 hours but every 1.5 hours. Thus, the process time (as part of the delivery time) has been reduced.

Step H. Repeat Step E for all Target EA alternatives	
Goal	To execute the measurements in the Target EA alternative models
Stakeholders	*IT Management*
Assumptions	• All important and participating concepts for the measurements can be found in the Target EA models designed in Step G or related views of the models. • The input data in the EA model are as more accurate as possible (considering that they are estimations/expectations).
Input	**Output**
• Measure Specification tables (Target EAs) • Target EA model alternatives • [Target EA analysis models] • Input data for calculating the measures • ‘Scale’ from Indicator Specification tables	• [Target EA analysis models] • Computed measures for the baseline architecture
Techniques	
• Any possible measurement approach (see Step E)	

Table 56 - Overview of Step H

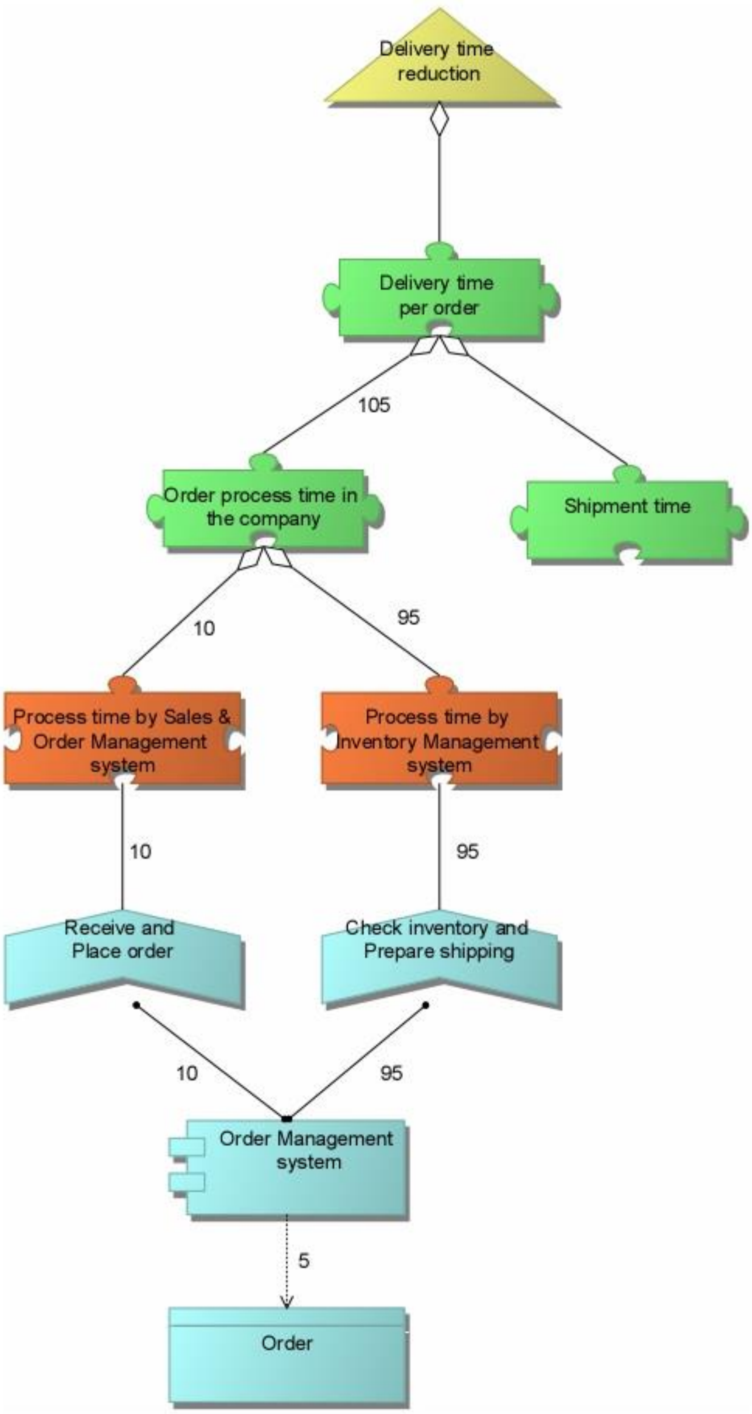
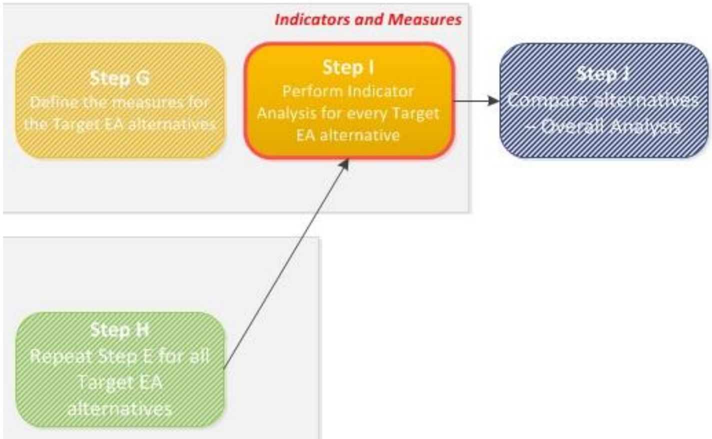


Figure 70 – Step F example: Target EA analysis model and measurement propagation

Step I: Perform Indicator Analysis for every Target EA alternative

Next in the process of goal quantification is to perform the actual analysis of the indicators which represent the goals. After acquiring the values for the measures in both the baseline and the target architecture alternative models, it is the turn of the value propagation across the PI structure to be realized.



The propagation will be based on the calculation formulas of the indicator specifications defined in Step C. Then, the assessment of the indicator values according to the decision criteria and interpretation criteria (analysis-specific attributes, Table 42) will be realized. The performance levels of the indicators determined through this process will finally lead to the goal satisfaction values. The three primary activities of Step I are presented in Figure 71 as well as the main deliverables of this step which are the indicator values, their performance levels and the goal satisfaction values.

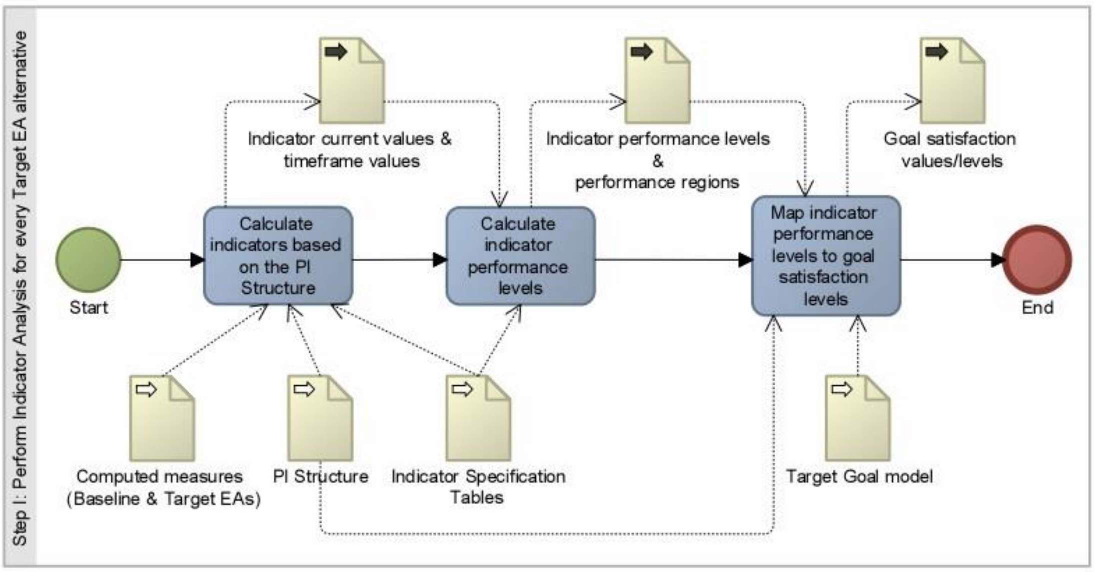


Figure 71 - Step I: Activities, Input & Output data

Activity I1. Calculate indicators based on the PI structure

The PI structure built in Step D is the guideline that will lead the value propagation for calculating the indicator values. By starting from the bottom of the PI structure, where the measure values reside, and following the calculation formulas for each indicator (determined in the Indicator Specification tables, Activity C2) the calculation of all the indicator values will be realized. Since the selection of the formulas and the models used has already been determined, the calculations performed in this activity are quite straightforward.

What is worth to put emphasis on is the explanation of the calculated value and what it stands for. The analysis model used for measuring an indicator entails a set of values and decision criteria. The value types that are important for its evaluation are its current value, the target value, the threshold value and the worst value, as proposed by the BIM approach in Figure 42. While the target, threshold and worst values have been determined in the Indicator Specification tables, the *current value* is yet to be assigned to every indicator. For every target EA model, there is a corresponding current indicator value as a result of the calculations in the propagation process. To put it differently, every *pair of baseline-target architecture* will result in a different *current value* to the indicator.

To facilitate the analysis and interpretation of the indicators' performance in Activity I2 as well as in Step J, a table stating the current value of each indicator for a specific target EA alternative model can be created (Table 57).

	Current values		
Indicator	*Target EA 1*	*Target EA 2*	*Target EA 3*
Indicator_name	Current value 1	Current value 2	Current value 3

Table 57 - Current values of Indicators

In addition, the specification of the timeframe of the indicator is determined in this activity. The timeframe of each indicator depends on the expected timeframe of the examined alternative, which results in multiple values depending on the target EA planning. The determination of the timeframe will assist in calculating the performance level of the indicator in the next activity. As it is suggested in Table 42, the time difference between the goal's timeframe and the indicator's (Ea alternative's) timeframe can serve as multiplier for adjusting the performance of the indicator and consequently the alternative. Additionally, when the indicators refer to monetary values, attention should be paid on the time difference between the baseline and the target architecture. Techniques which take into consideration the time value of money, i.e. dynamic measures, should be used (if they have not been already used will performing the measurements in Step H).

For example, for the indicator 'Delivery time reduction', the current value is 2550 minutes (1.8 days) as illustrated in Figure 72. This value represents the achieved reduction through the integration of the management systems as well as due to changes regarding the shipping process. In case a second target EA model had been designed for addressing the business goals, the indicator would end up with two different current values (one for each alternative).

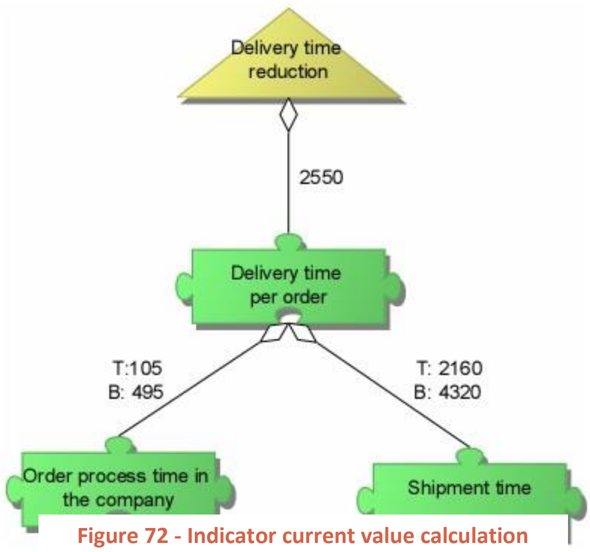


Figure 72 - Indicator current value calculation

<table>
<tr><th colspan="2">Activity I1. Calculate indicators based on the PI structure</th></tr>
<tr><td>Goal</td><td>To assign current values to each indicator through value propagation across the PI Structure</td></tr>
<tr><td>Stakeholders</td><td>IT Management</td></tr>
<tr><td>Assumptions</td><td>• The measurements are complete.
• All previous steps have been completed successfully.</td></tr>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>• PI Structure
• Indicator Specification Tables, especially the calculation formula property
• Computed measures (Baseline and Target Architectures)</td><td>• Indicator current values table
• Indicator timeframe values</td></tr>
<tr><th colspan="2">Techniques</th></tr>
<tr><td colspan="2">• PI Structures
• BIM Approach
• Financial Analysis Techniques: Dynamic measures</td></tr>
</table>

Table 58 - Overview of Activity I1

Activity I2. Calculate indicator performance levels

In Activity I2, the determination of the performance level of each indicator regarding a specific target architecture alternative takes place. For the calculation of the performance levels, the performance level equation introduced in Chapter 4 will be used. The equation is repeated below.

KPI Performance Level Equation

$$pl = \begin{cases} \dfrac{|current\ v. - threshold\ v.|}{|target\ v. - threshold\ v.|}, & if\ current\ v. \geq threshold\ v. \\ \\ -\dfrac{|current\ v. - threshold\ v.|}{|thershold\ v. - worst\ v.|}, & if\ current\ v. \leq threshold\ v. \end{cases}$$

As it can be derived from the equations above, the indicator properties needed for the calculations are the: *current value*, *target value*, *threshold value, worst value.* In addition to these values, the *performance levels & interpretation* property of the indicator will be also used for evaluating the performance level. Thus, the input for performing this activity includes the Indicator Specification Tables and the Indicator current values table. As it is proposed in the Indicator Specification table description, the performance level of an indicator can be influenced by the timeframe of the examined alternative. Considering the timeframe of the architecture solution and incorporating it in the evaluation of the performance level of the indicator comprises a more complete evaluation of the solutions regarding the corresponding goal's satisfaction. The Goal Specification table should be used for getting the goals' timeframe value. The formula that is proposed for calculating the final value of the performance level is repeated below for convenience:

$$time\_difference\ (td) = \frac{t_i - t_g}{t_g}$$

where: t_i is the timeframe of the indicator
t_g is the timeframe of the goal

Then, the *final performance level* is:

$$final\ pl = pl - pl * td$$

so, if $td > 0 \Rightarrow final\ pl < pl$

if $td < 0 \Rightarrow final\ pl > pl$

The output of the performance level value calculations can be formulated in a table stating the performance region to which the indicator belongs for every target architecture alternative (Table 59). Such a table can provide documentation for performing the comparison or trade-off analysis in the last step of the model, where the decision about which alternative should be chosen takes place.

	Performance level values & Performance regions		
Indicator	*Target EA 1*	*Target EA 2*	*Target EA 3*
Indicator_name	• pl_1 • interpretation/region	• pl_2 • interpretation/region	• pl_3 • interpretation/region

Table 59 - Indicator Performance level values & Performance regions

For example, if the analysis-specific attributes of the indicator 'Delivery time reduction' are the following:

Analysis-specific attributes for 'Delivery time reduction' indicator	
Current value	2550 minutes = 1.8 days
Target value $\rightarrow$ pl scale = 1	2 days = 2880 minutes
Threshold value $\rightarrow$ pl scale = 0	1.5 days = 2160 minutes
Worst value $\rightarrow$ pl scale = -1	0.5 days = 720 minutes
Timeframe	t_i = 10 months t_g = 12 months $time_{difference}\ (td) = \frac{t_i - t_g}{t_g} = -\frac{1}{6}$
Performance level – Interpretations	🔴 [-1, 0] : not worthy 🟠 [0, 0.5] : progressing 🟡 [0.5, 0.8] : very good 🟢 [0.8, 1] : best

By taking into account the values in the table above, the performance level of the indicator for this target EA model is:

$$pl = \frac{|current\ v. - threshold\ v.|}{|target\ v. - threshold\ v.|} = \frac{1.8 - 1.5}{2 - 1.5} = \mathbf{0.6}$$

$$final\ pl = pl - pl * td = \mathbf{0.7}$$

The performance level 0.7 falls into the performance region [0.5, 0.8] and, hence, the EA solution is considered 'very good' { 🟡 }. The coloring of the indicators could be applied on the PI structure as an indication of the performance of each indicator.

Activity I2. Calculate indicator performance levels	
Goal	To determine the performance levels and the performance regions of the indicators per target architecture alternative
Stakeholders	*IT Management*
Input	**Output**
• Indicator Specification Tables • Indicator current values table • Goal Specification table	• Indicator performance levels and performance regions
Techniques	
• BIM Approach	

Table 60 - Overview of Activity I2

Activity I3. Map indicator performance levels to goal satisfaction levels

Since the performance levels are determined, their assignment as goal satisfaction values is direct (Figure 73). Thus, the goal satisfaction quantification has been achieved by quantifying the actual improvements offered by the target architecture alternatives and transforming them into goal assessment values.

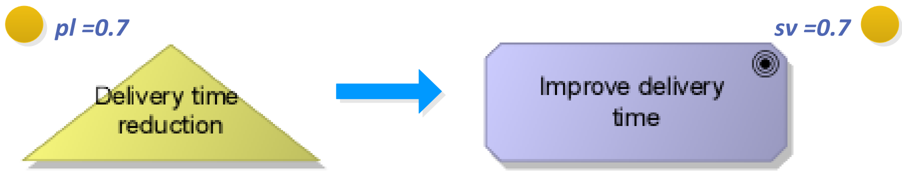


Figure 73 - Assigning satisfaction value (sv) to goals: example

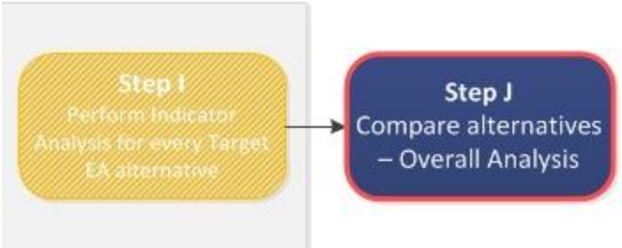
A simplistic approach has been followed throughout the EAGQ method, as it is assumed that, for every goal, only one indicator has been defined. In case more indicators are used, then the use of weights provides a method for calculating the satisfaction value of the goal. Introducing weights for aggregating the values of more than one indicators expressing one goal would mean that the strength of some indicators is greater than the strength of others. If weights are not applicable or do not reflect the relationship among the indicators characterizing the same goal, then the Range Normalization technique could be deployed for aggregating the performance level values of the indicators. The Range Normalization technique is part of the BIM approach.

The mapping of goal satisfaction values to indicator performance levels and the way the performance levels are calculated can be considered similar to the 'Goal Satisfaction Value Propagation based on KPI' approach.

Activity I3. Map indicator performance levels to goal satisfaction levels	
Goal	To assign the indicator performance level values to goals for representing the goals' satisfaction values
Stakeholders	*IT Management*
Assumptions	• Every goal is expressed through indicator(s) in the PI Structure.
Input	**Output**
• Indicator performance levels and performance regions • Target Goal model • PI Structure	• Goal satisfaction values/levels
Techniques	
• BIM approach: Range Normalization technique • Goal Satisfaction Value Propagation based on KPI	

Table 61 - Overview of Activity I3

Step J: Compare alternatives – Overall Analysis



The purpose of the final step of the EAGQ method is to suggest which the best target EA alternative solution is and to provide the visualizations of the comparisons. Step J proposes ways to perform the overall analysis of the change under examination. The assessment, comparison or evaluation of the possible architecture solutions that address the target goals can be based either only on the quantifications that are the result of the previous steps or it can also be combined with business case analysis techniques regarding, for instance, cost of investment and risks analysis, or with trade-off analysis techniques. Thus, the overall analysis can be customized on the needs of the user and can reflect the complexity of the change being investigated.

The first two activities, described below (Figure 74), comprise the basic analysis of the indicators and goal satisfaction values which is based on the output data of the EAGQ method. These two activities can provide a valuable outcome due to the quantifications. That is why Step J can be finished after the second activity has been completed.

The last activity concerns more complex methodologies and recommendations for further exploitation of the EAGQ method. Performing this activity is up to the user; hence, there could be a sequential relation between Activity J2 and J3, but it is not mandatory.

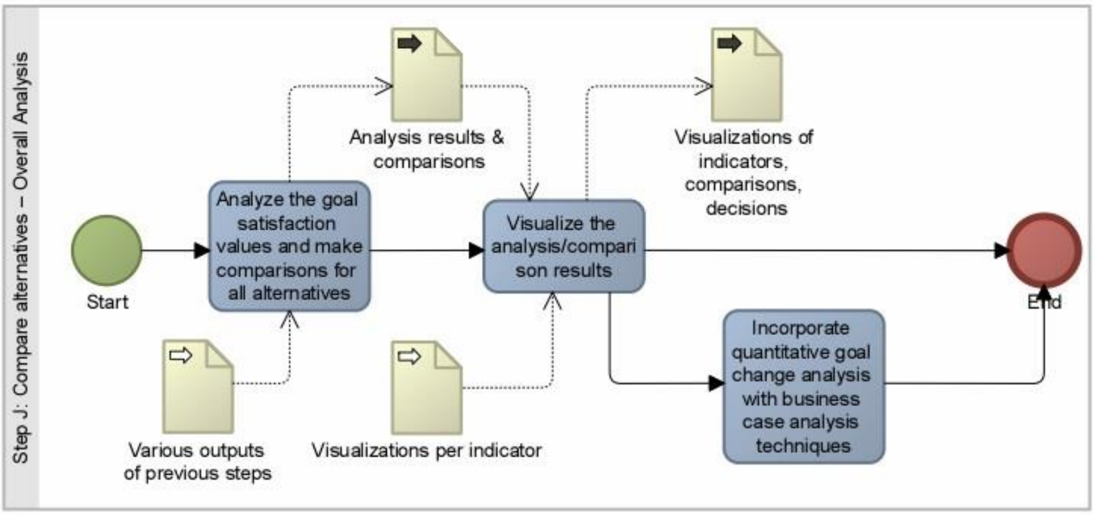


Figure 74 - Step J: Activities, Input & Output data

Activity J1. Analyze the goal satisfaction values and make comparisons for all alternatives

Deciding on the most suitable target architecture should be facilitated by analysis on the quantifications of the indicators and consequently the goal satisfaction values. There are various ways to utilize the output of the value propagation and the performance levels identified for each indicator. Combining the generated information appropriately will result in the most desirable and beneficial outcome.

Information that can be used for analyzing the quantifications are the *probability* assigned to each indicator, its *timeframe* (depending on the alternative) as well as the *visualizations* proposed for every indicator. For instance, the probability of the each alternative solution to provide the expected improvements combined with its elements' contribution to the goals or indicators comprise a way of assessing the architecture alternatives. How well each alternative performs comparing to the baseline architecture can be evaluated by simply observing the indicator performance levels by themselves; while comparisons among alternatives entail the identification of the alternative which provides better results under certain circumstances (e.g. time). When an investment decision is examined, making no changes is also an option. Situations where the baseline architecture provides more benefits can be identified when the evaluation type of the goal is not aligned with the outcome of the indicator or when the performance region of the majority of the indicators is very low. Such observations denote that the quality of the target architecture(s) is low and that the baseline architecture is more suitable.

Combining various pieces of information used throughout or produced by the EAGQ method, the satisfaction values of each goal and the degree of contribution of the architectures is a form of trade-off analysis, where multiple criteria are considered and examined for making a decision.

An example method for EA-scenario analysis based on quality attributes was proposed by Davoudi et al. [117] in 2009. This approach comprises a six-step method which uses the Analytical Hierarchy Process (AHP) for prioritizing quality attributes and assesses different EA-scenarios according to these quality attributes. The outcome of the approach is, firstly, the solution EA-scenario which best fits the list of prioritized quality attributes and, secondly, the level of uncertainty that the selection of this solution entails. By viewing or treating goals (issue or focus of the goal) as the quality attributes and the indicators or measures as the criteria of the attributes (used for assessing the attributes); this method could be applied in this activity for evaluating and comparing the target EA alternatives based on the quantified indicators and measures. The topic of goal prioritization has not been addressed in this thesis, since it is considered out of scope, but such methods could facilitate the overall analysis and the decision-making process. More goal and requirement prioritization techniques have been discussed, for instance, by Karlsson and Ryan [10], Wiegers [118], Berander and Andrews [119].

The results of the analysis can be of any type, e.g. documentations, tables, etc. In case visualizations are needed, the next activity is responsible for dealing with them. For supporting visualizations, decisions on what type of comparisons and output are expected and what type of data are to be used as input of the diagrams, pies, charts, etc. can comprise the input for Activity J2.

<table>
<tr><th colspan="3">Activity J1. Analyze the goal satisfaction values and make comparisons for all alternatives</th></tr>
<tr><td>Goal</td><td colspan="2">To compare and evaluate the architecture alternatives in terms of the baseline architecture and in terms of their contribution to the goal change.</td></tr>
<tr><td>Stakeholders</td><td colspan="2">IT Management and Boardroom</td></tr>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td colspan="2">• Indicator performance levels and performance regions
• Goal satisfaction values/levels
• Indicator Specification Tables
• Measure Specification Tables
• EA models
• Target Goal model
• Computed measures (Baseline and Target Architectures)
• PI Structure</td><td>• Analysis results and comparisons
• Proposals for visualizations</td></tr>
<tr><th colspan="3">Techniques</th></tr>
<tr><td colspan="3">• No specific techniques are used in this activity except from simple comparisons and contribution evaluations.
• Priyanto's EA-based Decision Making Method (for considering the visualization options)</td></tr>
</table>

Table 62 - Overview of Activity J1

Activity J2. Visualize the analysis/comparison results

When the comparisons have been decided or performed, their visualization can enhance the understanding of the stakeholders and can improve the decision-making process. Except from the visualizations of the indicators determined in Activity C3 and the proposals in the previous activity, the determination of the most suitable visualizations of indicators to show the essence of the outcome of the decision taken is also important.

<table>
<tr><th colspan="3">Activity J2. Visualize the analysis/comparison results</th></tr>
<tr><td>Goal</td><td colspan="2">To provide support to decision-making through visualizations of the comparisons and the decisions.</td></tr>
<tr><td>Stakeholders</td><td colspan="2">IT Management and Boardroom</td></tr>
<tr><th colspan="2">Input</th><th>Output</th></tr>
<tr><td colspan="2">• Visualizations per indicator
• Analysis results and comparisons
• Proposals for visualizations</td><td>• Visualizations of indicators, comparisons, decisions</td></tr>
<tr><th colspan="3">Techniques</th></tr>
<tr><td colspan="3">• Priyanto's EA-based Decision Making Method</td></tr>
</table>

Table 63 - Overview of Activity J2

Activity J3. Incorporate quantitative goal change analysis with business case analysis techniques

The comparisons performed throughout this model are based on one primary rule, i.e. the implementation and migration period of the project has been completed; hence, the alternative, when examined, is fully functional, like the current architecture. This rule serves as a basis for being able to compare two architectures under the same terms/conditions. Consequently, the result of such comparison is the expected benefit from deploying the target architecture. For making an investment decision, though, more factors play role.

Firstly, the consideration of the *investment budget* is highly important. An assessment of the costs for the acquisition of the new systems, the implementation and in general the transformation from the baseline architecture to a target one has to be included. Techniques that can support such analysis are financial analysis techniques, such as ROI, performance index, CBAM or the cost-benefit EA analysis method. The last two can be considered as trade-off analysis techniques as well, since they include quality attributes and uncertainty for the assessment of the decision.

Another important factor, which can be combined with the investment and implementation costs, is the time-span of the investment. This factor coincides with the timeframe of the alternative. While time has already been used in the determination of the performance level of the indicators, in this level of analysis time-span can facilitate techniques such as the payback period or break-even analysis.

Finally, for a business case to be complete, risk analysis is essential. Such analysis is not in the scope of this thesis or the EAGQ method. The determination of the probability of the indicator (benefit) to be realized, though, can support such analysis.

<table>
<tr><th colspan="4">Activity J3. Incorporate quantitative goal change analysis with business case analysis techniques</th></tr>
<tr><td>Goal</td><td colspan="3">To provide a better supported decision-making process by incorporating the quantitative goal analysis with business case analysis techniques or aspects.</td></tr>
<tr><td>Stakeholders</td><td colspan="3">IT Management and Boardroom</td></tr>
<tr><th colspan="2">Input</th><th colspan="2">Output</th></tr>
<tr><td colspan="2">• Quantitative analysis data and visualizations</td><td colspan="2">• Business case and trade-off analysis</td></tr>
<tr><th colspan="4">Techniques</th></tr>
<tr><td colspan="4">• Financial Analysis Techniques</td></tr>
</table>

Table 64 - Overview of Activity J3

CHAPTER 6

6. Demonstration of the EAGQ Method

The goal of this chapter is to prove that the EAGQ method can be applied in practice by demonstrating the method through the ArchiSurance case study. The ArchiSurance case study is a fictitious example which was developed by Jonkers et al. [120] for demonstrating the use of ArchiMate® language and TOGAF framework, and it was published by The Open Group. The ArchiSurance organization is a result of a merger of three previously independent companies and due to existing challenges, change scenarios are being considered. The ArchiSurance case comprises a case of medium complexity and provides a good setting for demonstrating the method developed in this thesis. The existing challenges provide the motivation that leads the change, while the baseline architecture is also described in detail and, thus, offering a solid background of the case and the current situation. A few modifications have been made in the case for facilitating the demonstration process.

The focus of the ArchiSurance case study is mainly on costs, thus a financial and cost analysis is the core of the demonstration. Since the EAGQ method can also support performance measures' analysis, a few time-related measures have also been selected to illustrate this capability of the method. In the next sections, first, an introduction to the case study is provided and, then, the application of the EAGQ method is described. In the end of the chapter, an example algorithm in pseudocode is provided for demonstrating the feasibility of the EAGQ method's indicator value propagation to be implemented.

6.1. Background and Current situation of the ArchiSurance organization

ArchiSurance is an insurance company which was formed after the merger of three previously independent insurance companies, namely: (i) Home & Away, (ii) PRO-FIT, and (iii) Legally Yours, each of which has an expertise in a different insurance domain (Figure 75). The main drivers for realizing the merger, in the first place, were the opportunities for better cost control, maintenance of customer satisfaction, new technology investments and taking advantage of emerging markets, since the competitiveness in their markets was increasing against them.

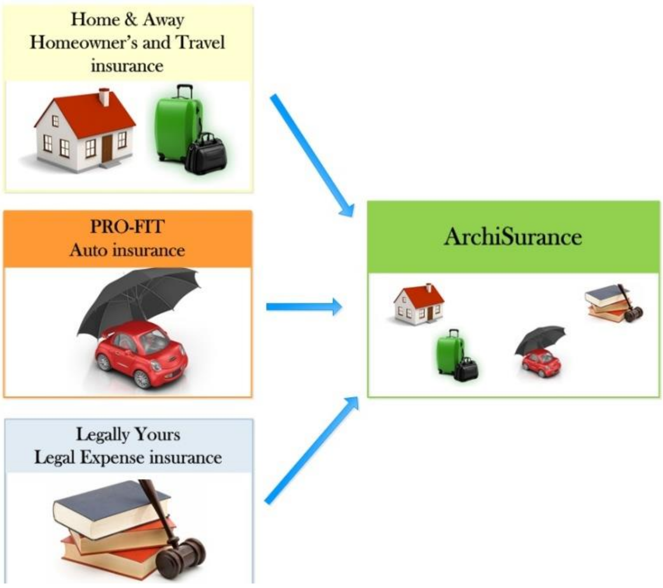


Figure 75 – ArchiSurance case study: Merger of three independent companies

After the merger, all three pre-merger companies continue to exist in the new company, i.e. the ArchiSurance, as three separate divisions. Additionally, all insurance products, previously offered by the three pre-merger companies, are still offered by ArchiSurance.

First steps have been made towards the integration and alignment of the three companies. Firstly, the setup of the shared front-office aimed to provide a multi-channel contact center for sales and customer service through a common web portal and a contact center software suite. Secondly, the integration was further facilitated by the establishment of a Shared Service Center (SSC) for document processing, which, among others, has assisted the automation of the document workflows. A third initiative is related to the CRM system. The company has implemented a strategic CRM solution for both Home & Away and PRO-FIT, but due to concerns regarding post-merger risks and performance issues, Legally Yours still uses its own CRM system. In addition to that, the existence of three separate back-offices and the use of the pre-merger individual applications and systems cause problems in handling the insurance products in the post-merger era. The complexity and redundancy of the current (baseline) application landscape can be observed in Figure 76.

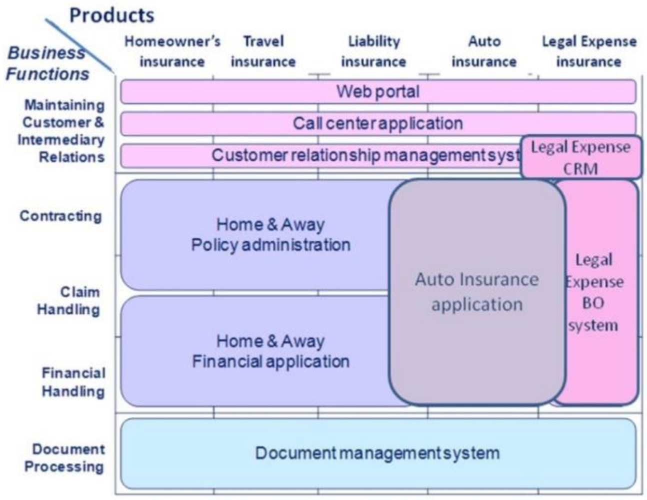


Figure 76 – Baseline Application Landscape

ArchiSurance's decisions and goals are based on a set of drivers, such as profit, price, customer satisfaction which serve as guidelines for determining business goals. While some of the goals have been already addressed by the first phase of the merger, the company faces challenges in the current landscape that need to be dealt with. Among the main remaining goals are the achievement of IT cost savings and the increase of business adaptability to changes. The problems that have been identified as obstacles to these goals are the scattered application landscape and the resulting data redundancy and functional overlap, as well as the point-to-point application integration which necessitates the use of a variety of data formats and methods [120].

The acknowledgement of the problems in relation to the drivers and business goals has lead to the determination of two change scenarios. The focus of the ArchiSurance case study is upon the 'Application Portfolio Rationalization' change scenario, which is going to be the focus of the demonstration of the EA-based Goal Quantification Method as well. Based on the information provided in the ArchiSurance case study by Jonkers et al. [120] and the documentation of the baseline architecture, a goal quantification analysis is going to be performed for evaluating target architecture alternatives that can realize the target goals and the change scenario. While the analysis is considered cost- and financial-oriented, performance issues are also addressed.

6.2. Application of the EAGQ Method on ArchiSurance

Based on the description of the ArchiSuarance background and current situation, a set of goals are identified. The set of goals mentioned in the case study [120] are incomplete that is why for demonstrating the method, more goals have been defined as well as requirements realizing the goals.

Step A: Build the Target Motivation Model

Activity A1. Build the Goal Model

The main driver that leads the transformation of ArchiSurance organization is the profit. After the first post-merger expectations being met, an assessment of the profitability of ArchiSurance indicated that there are increased application maintenance costs, high personnel costs due to obstacles of sharing information across the company as well as relatively low cross-selling of insurance policy types due to scattered front-office customer data access resulting from separate CRM systems. These assessments led ArchiSurance to define new goals for improving this situation and enhancing the integration and alignment among the three divisions as well as in the front-office of the organization.

As it was proposed in the description of Step A in the previous chapter, the goal model design can be supported by the first phase of the Goal-driven Measurement Process (GQ(I)M) and/or the GQM+ Strategies approach. The GQM+ Strategies approach is rather interesting in the ArchiSurance case. Considering that lower-level goals are derived from the higher-level business goals through the identification of strategies, the results of the assessment regarding maintenance costs and information sharing can support the selection of strategies for forming lower-level goals.

The 'Application Portfolio Rationalization' change scenario, which is the basis of this demonstration, specifies the solution in detail [120]. Since one of the steps of the EAGQ method (Step F) is to define target EA alternatives, the detailed specifications are going to be avoided in this step and only the two main points of the scenario are going to be used [120]:

1. Migrating to an *integrated back-office suite* for the functions of policy administration and financial transactions.
2. Completing the migration to the *strategic CRM system*.

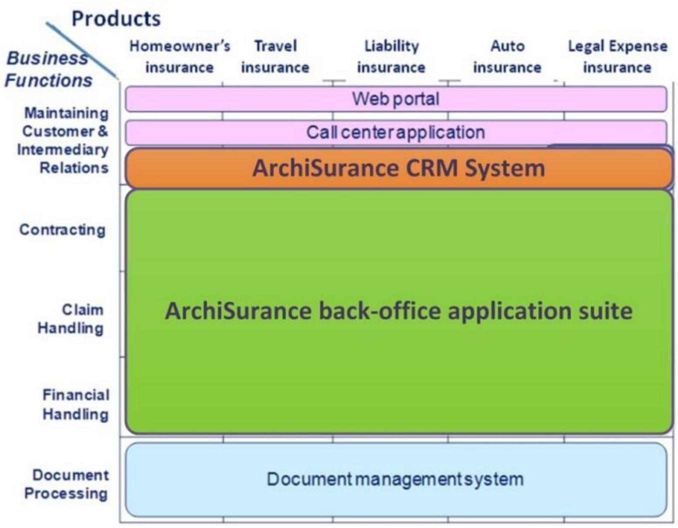


Figure 77 – Target Application Landscape

The target application landscape of the ArchiSurance organization is presented in Figure 77. The figure is redesigned based on a figure in the ArchiSurance case study [120]. The requirements in the Target Goal Model, presented in Figure 78, are in line with the change scenario, while the actual implementation of the target architecture is not yet revealed.

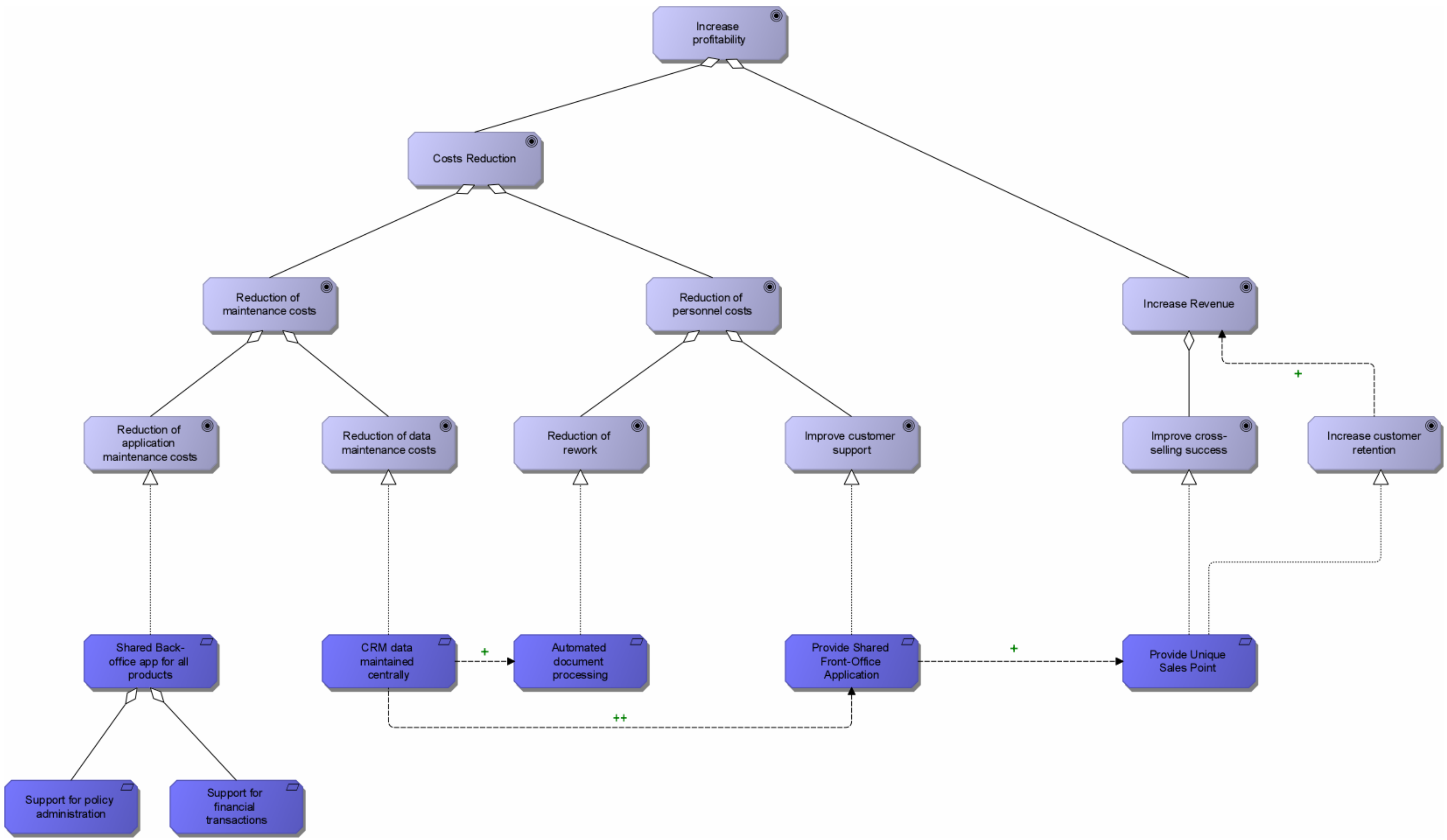


Figure 78 - ArchiSurance Target Goal Model

Activity A2. Define goal attributes

While determining the goals, specific attributes of the goals need to be defined as well. The most important properties are the 'target value' and the 'timeframe' for achieving the goal. We can assume that the two main points of the 'Application Portfolio Rationalization', i.e. integrated back-office suite and integrated CRM system, will have different realization deadlines. For the back-office suite, the timeframe is two years, since currently every division uses its pre-merger application, while for the CRM system the desired timeframe is one year. Part of the integration of the CRM system has already been achieved through the integration of the CRM systems of Home & Away and PRO-FIT, which explains the shorter timeframe. The Goal Specification Table (Table 65) includes the attribute values for every goal. For the soft goals, the values are in the form of percentages, and for the hard goals the values in the form of absolute monetary values.

Goal Specification table					
	Attributes				
	Level	Measurability	Evaluation type	Target value	Timeframe or Deadline
Goal	Tree level (0 – n)	(S)oft or (H)ard	(A)chieve (C)ease (M)aintain A(v)oid (I)ncrease (D)ecrease		
Increase profitability	3	S	I	3%	2 years
Costs Reduction	2	H	D	212K €	2 years
Reduction of maintenance costs	1	H	D	190K €	2 years
Reduction of application maintenance costs	0	H	D	120K €	2 years
Reduction of data maintenance costs	0	H	D	70K €	1 year
Reduction of personnel costs	1	H	D	22K €	1 year
Reduction of rework	0	S	D	20%	1 year
Improve customer support	0	S	I	80%	1 year
Increase Revenue	1	S	I	30%	1 year
Improve cross-selling success	0	S	I	30%	1 year
Increase customer retention	0	S	I	20%	1 year

Table 65 - ArchiSurance Goal Specification table

Step B: Map the Target Goal Model to the Baseline EA model

Activity B1. Build the Baseline EA model

For building the Baseline EA model, information about all three domains should be gathered. For the business domain, the business functions and processes as well as their interaction are important for understanding how the company interacts with its customers as well as with other internal and external actors. The main functionality is expressed in the Baseline EA model in Figure 81 by the business services and their supporting business processes. As an insurance company, ArchiSurance offers a process for selling a new insurance product, for handling a claim which also includes the payment of the claim, providing information to the customer and collecting the premium from its customers depending on their insurance products. The actors and roles that are influenced by the change that will take place are the customer, the front-office, the back-office and possibly the people working at the SSC.

The application landscape illustrated in Figure 76 indicates the existing architecture in the application domain. A set of services are also defined for exposing and providing the functionality of the applications to the business layer. As it can be observed the density of the lines is quite high, since the same functionality is offered by different applications (Figure 80). The specialization of the applications to a specific pre-merger insurance company type is the reason for this complexity.

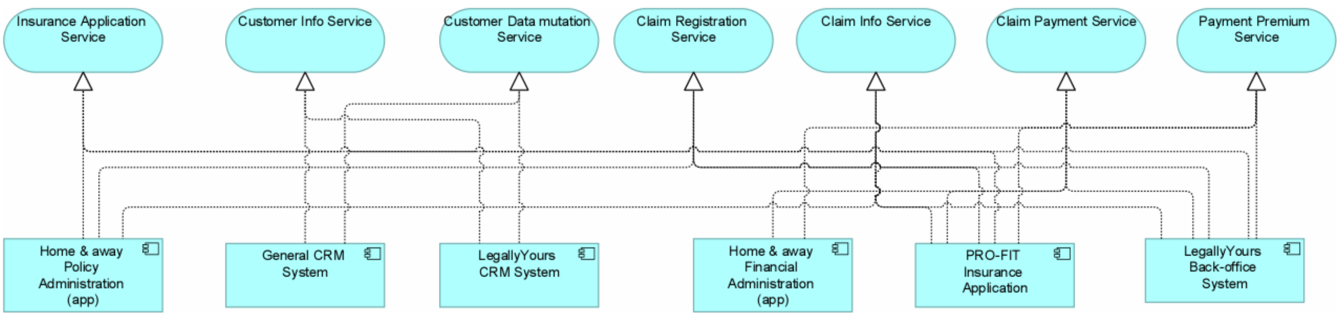


Figure 80 – Baseline Application domain complexity

Finally, the technology architecture is imposed by the locations of the three pre-merger companies' headquarters, which have remained in the locations of the prior independent companies. After the merger, the shared front-office and the SSC are the only additions made which were placed in the Home & Away and PRO-FIT headquarters respectively. Figure 79 demonstrates the locations of the servers of the ArchiSuarnce organization. Based on the provided information from specialists in the organization about the landscapes and the functionality offered by ArchiSurance, the Baseline EA model is designed in Figure 81.

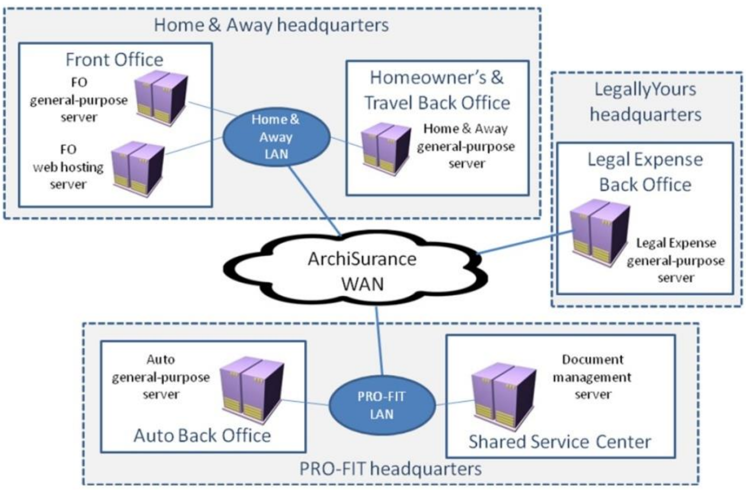


Figure 79 - Baseline Infrastructure Landscape, obtained from [120]

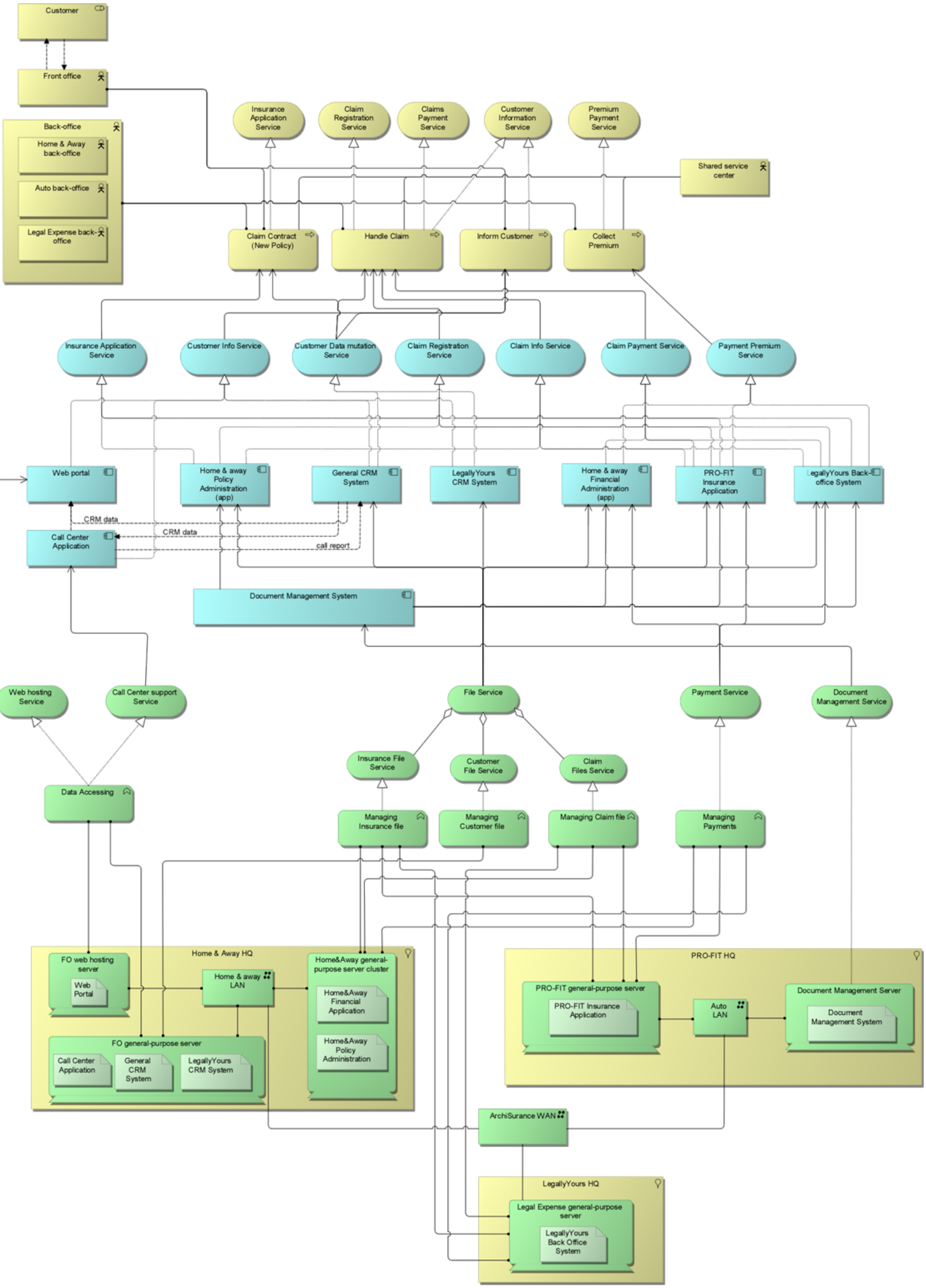


Figure 81 - ArchiSurance Baseline EA model

Activity B2. Relate the Baseline EA concepts with the Requirements

For deciding on what are the possible changes needed in the architecture regarding the new goals, AchiSurance has to understand how the baseline architecture realizes the target goals. Discussions among various stakeholders and experts have resulted in a set of relations among concepts of the baseline architecture and the target goal model. A partial view of the requirements' realization viewpoint which can be used for demonstrating the relationships is designed in Figure 82 for the 'Shared back-office app for all products' requirement. The complete view of the requirements' realization by the baseline architecture can be found in APPENDIX VIII.

Figure 82 indicates that the group of requirements which realize the goal 'Reduction of application maintenance costs' is realized by a number of services (realization relations) but the systems which should realize the services both in the Application domain and the Infrastructure domain are not suitable. Thus, the association relation is used for demonstrating what the source of the application maintenance costs is in the baseline architecture.

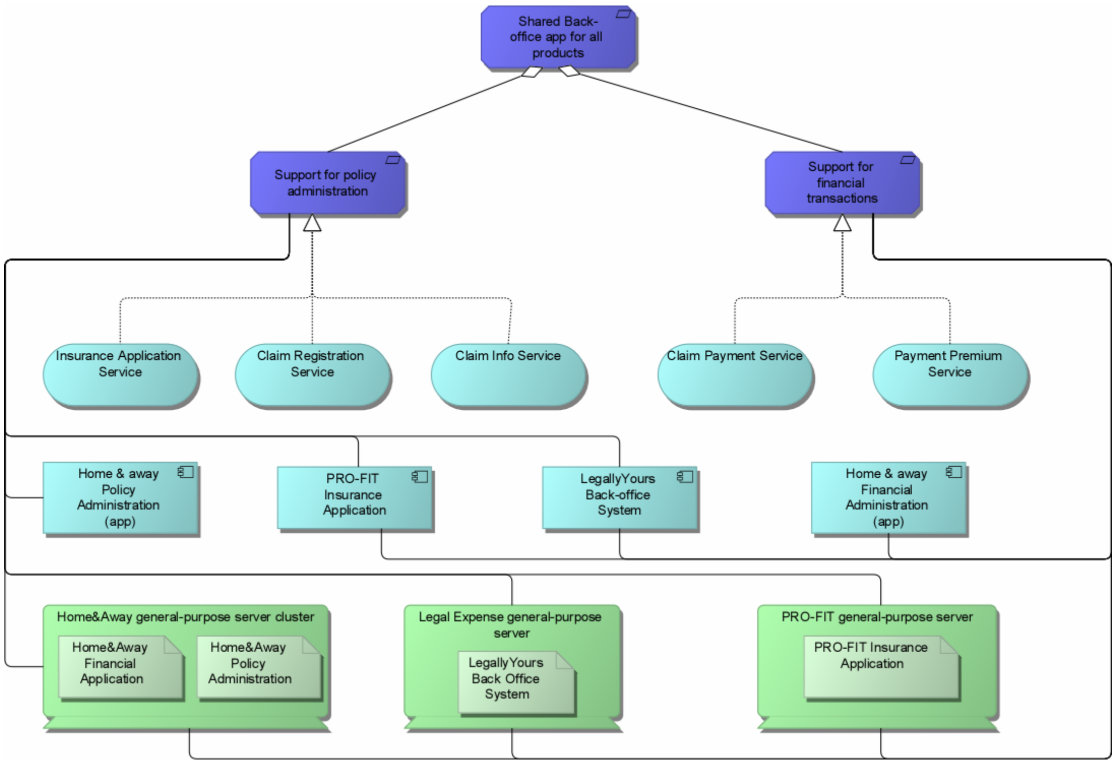


Figure 82 - 'Shared back-office app for all products' requirement realization in Baseline EA model

Activity B3. Identify problems/gaps in the realization of the Requirements

Based on the relation types used for showing the degree of realization of a requirement by a core concept, the requirements which are partly covered or not covered by the baseline architecture of the ArchiSurance organization are identified. The categorization of the requirements is documented in Table 66, while the different colors demonstrate the categorization in the model (*red* → *not covered, orange* → *partly covered, purple* → *covered*). In the complete requirements realization in APPENDIX VIII, the same colors are used for illustrating the differences among the requirements.

Requirement realization-level	Requirements	Gap estimation
Covered	Automated document processing Provide Unique Sales Point	• No potential gap • Requirement underperforming
Partly covered	Support for policy administration Support for financial transactions CRM data maintained centrally Provide Shared Front-Office Application	Potential gap: • Requirement underperforming • Requirement partly fulfilled
Not covered	Shared Back-office app for all products	Gap existence: • Requirement not realized at all

Table 66 - ArchiSurance requirements categorization

Step C: Identify indicators and measures

Activity C1. Ask questions for the lowest-level goals to reveal the measurement needs

In the Goal Specification table (Table 65), the level of the goals was determined. The goals with level-value equal to zero are the lowest-level goals (Figure 83). Additionally, in Figure 83, questions are provided for the goals 'Reduction of application maintenance costs' and 'Improve customer support' as examples. The selection of the two goals is based on the different approach taken for measuring them. Even if 'costs' is the primary purpose of the measurement, the first goal's questions refer to cost identification, while the second goal's questions refer to time.

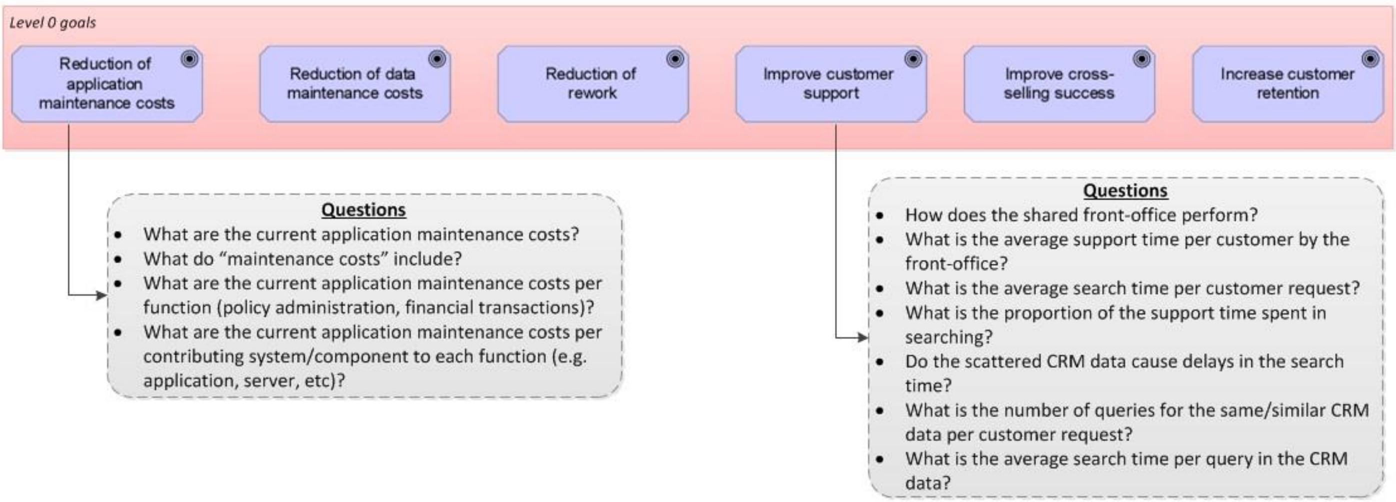
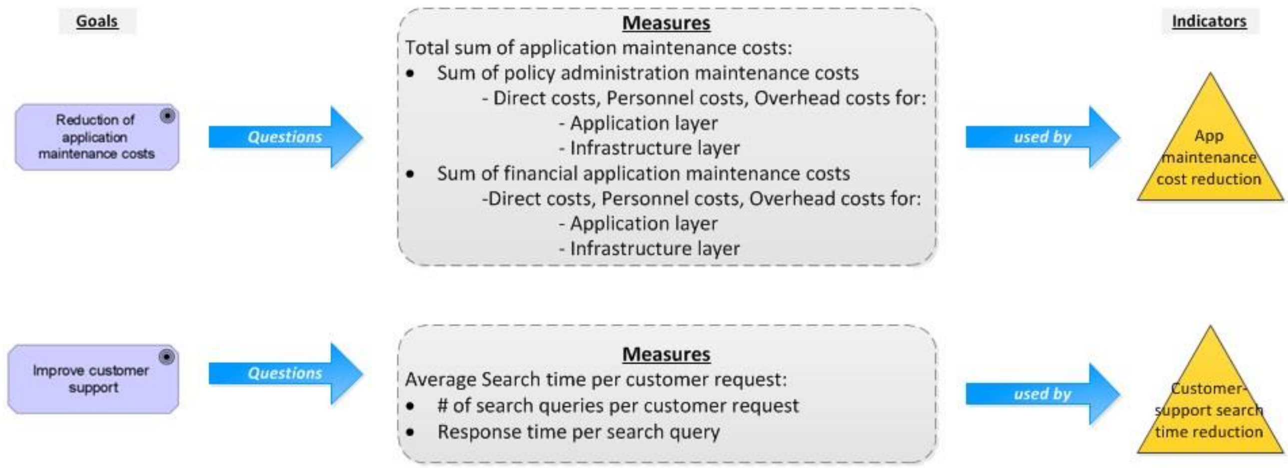


Figure 83 – ArchiSurance lowest-level Goals and example Questions

Activity C2. Assign indicators to each goal

Based on the questions for clarifying the measurement needs of the goals, the indicators and measures that are appropriate for quantifying the goals need to be established. Starting from the level-0 goals, the measures and indicators for these goals are decided by answering the questions. So, for the goals analyzed in Activity C1 (Figure 83), the measures and indicators are the following:



Then, the indicators for the higher-level goals are determined. By having in mind what is measured and what the indicators of the lower-level goals are, the specification of the upper-level indicators becomes clearer. The selection of the most appropriate and expressive indicator, though, will entail research in performance indicator repositories either of the organization or available in books, papers, websites, frameworks, theories, etc. Repeating Activity C1 for asking questions for the goals with no straightforward indicators is also an option, since it can assist to perform a more effective indicator investigation. Table 67 provides the indicators determined for the Target Goal model in Figure 78. Most of the indicators in the table are directly derived from the goals. The only indicators which appear to have a divergence from the corresponding goals are:

1. *Goal*: Reduction of rework → *Indicator*: Rework-time reduction
2. *Goal*: Improve customer support → *Indicator*: Customer-support search time reduction
3. *Goal*: Improve cross-selling success → *Indicator*: Average premium increase per customer
4. *Goal*: Increase customer retention → *Indicator*: Revenue retention rate increase

The selection of these indicators was made by having as a primary criterion the ability to measure the indicator in the architecture. Additionally, research has been conducted for identifying indicators which have been used in practice or in theory for measuring the corresponding goals. A discussion among ArchiSurance IT stakeholders has also taken place for pinpointing the most important issues regarding the goals' focus.

For facilitating the demonstration of the method, a third column has been added in the 'Goal-Indicator Pairs' table (Table 67) where an overview of the measures proposed for each lowest-level indicator is offered. In the APPENDIX VIII, information about the calculation formulas of each indicator and its corresponding measure is also provided.

Goal – Indicator Pairs		
Goal	**Indicator**	**Measures**
Increase profitability	Profitability Index	
Costs Reduction	Total Cost Reduction	
Reduction of maintenance costs	Maintenance cost reduction	
Reduction of application maintenance costs	App maintenance cost reduction	• Total sum of app maintenance costs ○ Sum of policy administration maintenance costs ○ Sum of financial application maintenance costs
Reduction of data maintenance costs	CRM maintenance cost reduction	• Sum of CRM data maintenance costs
Reduction of personnel costs	Personnel cost reduction	
Reduction of rework	Rework-time reduction	• Total rework time
Improve customer support	Customer-support search time reduction	• Average Search time per customer request
Increase Revenue	Achived revenue increase	
Improve cross-selling success	Cross Selling premium increase	• Total annual premium from customers with more than one contracts
Increase customer retention	Revenue retention rate increase	• Revenue retention rate

Table 67 - ArchiSurance Goal-Indicator pairs (plus measures)

The Indicator and Measure Specification tables are used for defining the indicators' and measures' attributes. To achieve a concise demonstration which will present the essence of the model, only one example for each specification table is provided. The Indicator

Specification table is used for determining the 'App maintenance cost reduction' indicator (Table 69) and the Measure Specification table for the 'Sum of policy administration maintenance costs' measure (Table 70). Thus, the main outputs of this activity are:

1. the identification of measures by answering the questions from Activity C1
2. the determination and assignment of indicators to goals (Table 67)
3. the specification of each indicator for defining its attributes, e.g. Table 69
4. the specification of each measure used for calculating the indicators, e.g. Table 70

For determining the monitoring objects that play a role in the measurement of the measures, an early application of Activity E1 will result in EA analysis models. An example Baseline EA analysis model is provided for the 'Sum of policy administration maintenance costs' measure in Figure 84.

Furthermore, the assignment of indicators to goals has resulted in reconsidering about the 'measurability' attribute of the goals in the model. The goals: 'Reduction of rework', 'Improve customer support', 'Increase Revenue' and 'Improve cross-selling success' were identified as 'soft' goals in the Goal specification table (Table 65). With the help of time measures and revenue measures, these goals have been turned into 'hard' goals and, consequently, their 'target value' attribute can also be refined after performing the measurements in the baseline EA model. For example, 20% of rework time reduction would be in absolute value: $20\% * B_{ReworkTime}$. Table 68 provides the updated fields of the ArchiSurance Goal Specification Table.

Goal Specification Table					
	Attributes				
	Level	Measurability	Evaluation type	Target value	Timeframe or Deadline
Goal	Tree level (0 – n)	(S)oft or (H)ard	(A)chieve (C)ease (M)aintain A(v)oid (I)ncrease (D)ecrease		
Reduction of rework	0	**H**	D	**20% * $B_{ReworkTime}$**	1 year
Improve customer support	0	**H**	I	**80% * $B_{SearchTime}$**	1 year
Increase Revenue	1	**H**	I	**30% * $B_{revenue}$**	1 year
Improve cross-selling success	0	**S/H**	I	**30% *or* 30% * $B_{CrossSellingpremium}$**	1 year

Table 68 – ArchiSurance Goal Specification table: Updated fields

Indicator Specification: "App maintenance cost reduction"	
Benefit	*Benefit Specification Matrix* **Classification change** | **Measurement of effect** | **Probability** Do things better | Financial | 90%
Scale	Money in €
Measures	Total sum of app maintenance costs • $\mathbf{T_{app}}$ for the target EA • $\mathbf{B_{app}}$ for the baseline EA
Calculation formula	$CR_{app} = B_{app} - T_{app}$ where B_{app} : *Baseline* app maintenance costs T_{app} : *Target* app maintenance costs
Target value → pl scale = 1	Reduction by **120K €**
Planned values	Reduction by **70K €** after the 1^{st} year
Threshold value → pl scale = 0	**85K €**
Tolerance value	**2K €**
Worst value → pl scale = -1	**60K €**
Performance level – Interpretation	[-1, 0] : not worthy [0, 0.2] : basic/minimum [0.2, 0.5] : progressing [0.5, 0.8] : very good [0.8, 1] : best
Timeframe	*Defined in Activity I1*
Frequency of measurement	recurring → maintenance costs calculated *per year based on monthly costs*
EA layer	Application layer & Infrastructure layer
Visualization	*Defined in Activity C3*

Table 69 - 'App maintenance cost reduction' indicator specification table

Measure specification: "Sum of policy administration maintenance costs"	
B$_{policy}$: *Baseline* policy administration app maintenance costs	
Type of measure	Sum of costs (€)

Component measures	From Table 75, the application maintenance costs include:
	• s/w maintenance costs
	  ○ **upgrade costs**
	  ○ **license fees**
	• h/w maintenance costs
	  ○ **s/w update costs** (s/w running on servers)
	  ○ **h/w upgrade costs** due to new needs (see Ongoing Change & Growth costs in Table 75)
	• personnel costs (#people, hours, salary): **s/w & h/w maintenance labor costs**
	• overhead/indirect/hidden costs comprising of:
	  ○ **downtime costs**
	  ○ **documentation costs**
	  ○ **administrative costs**
	  ○ **depreciation costs** (see Disposal costs in Table 75)

Type of scope	*Temporal*: data after the merger
Type of monitoring objects and points	The monitoring objects are presented in Figure 84 (as a result of Activity E1).

Table 70 - 'Sum of policy administration maintenance costs' measure specification table

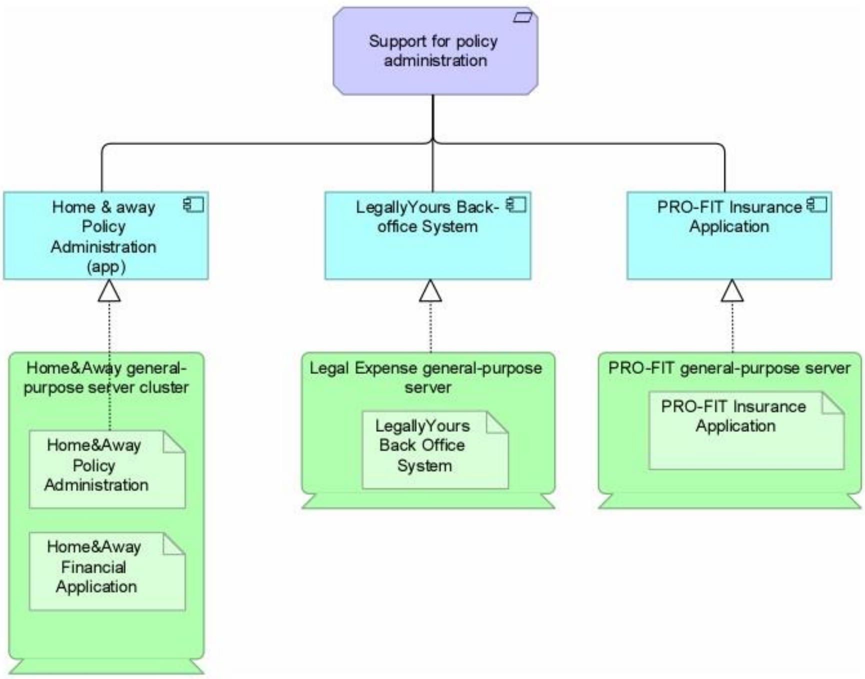


Figure 84 – 'Sum of policy administration maintenance costs' measure's monitoring objects

Activity C3. Define visualizations for the indicators

Based on Priyanto's framework and by considering the data types expected from the measurements, the most expressive visualizations are defined for each indicator. In Table 71, the visualizations selected for the 'App maintenance cost reduction' indicator is presented along with the context of the visualization.

Visualizations for indicator: "App maintenance cost reduction"	
Visualization type	**What to represent**
Column chart	• Baseline app maintenance costs • Each target EA app maintenance costs
Pie chart	• Portion of total cost reduction that the 'app maintenance cost reduction' represents

Table 71 - Visualization types for the 'App maintenance cost reduction' indicator

Activity C4. Determine the Focus area of each goal in the Baseline architecture

The focus area of each goal is determined based on the information provided mainly from the measures specifications and the relations between baseline EA concepts and requirements identified in Activity B2. Additionally, the Target goal model will help in guiding the assignment of core concepts to higher-level goals.

For example, the focus area of the goal 'Reduction of maintenance costs' is provided in Figure 85. This view comprises a combination of the 'Goal Realization Viewpoint' and the 'Requirements Realization Viewpoint' defined in ArchiMate 2.1 Specification [17]. By combining the two viewpoints, the visualization of the traceability links between the goals and the EA concepts is enabled. There is a difference though. In the realization viewpoints only the realization relations are taken into account. Regarding the focus area graphs, the association relations among requirements and concepts of the architecture model are also considered.

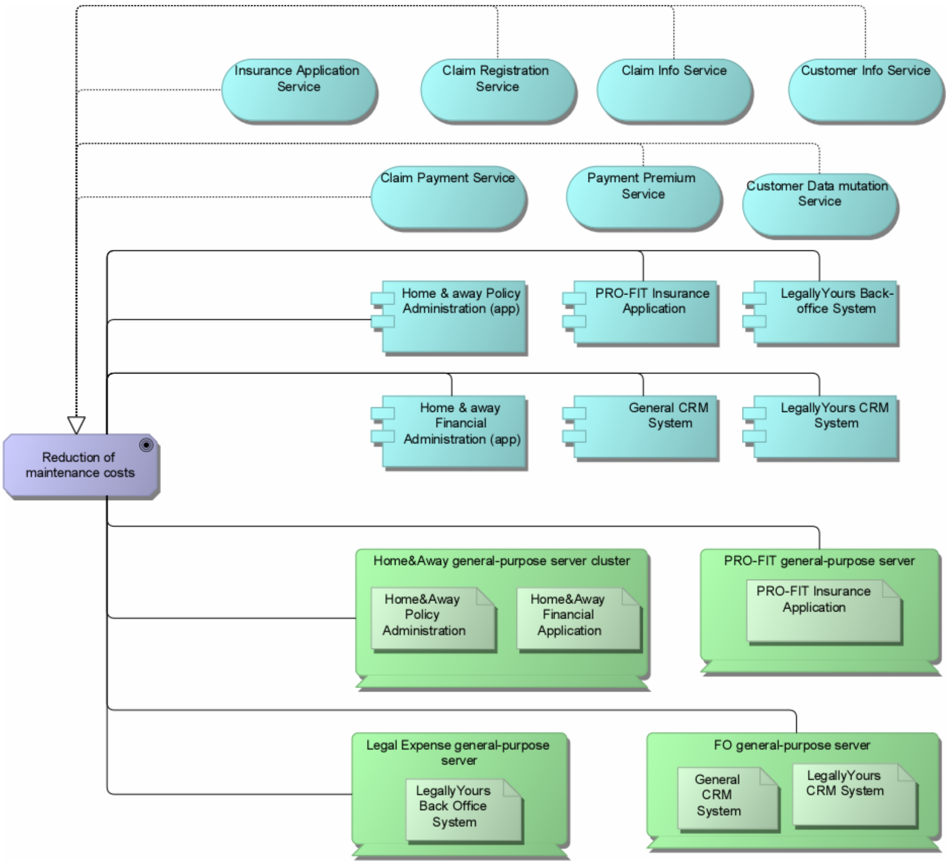


Figure 85 - Focus area of 'Reduction of maintenance costs' goal

Step D: Build the PI structure

Activity D1. Examine for duplicate measures in the model

This activity is necessary when limited attention has been paid during the definition of the indicators and measures in the previous step. In the ArchiSurance case, the specified measures may seem unique, but in reality there is some redundancy.

The redundancy is a result of the *overlap* between the *policy administration* and *financial application maintenance costs* for the applications of PRO-FIT and Legally Yours. The applications of these divisions provide both functionalities; thus, when the component measures for calculating the total policy administration and financial app maintenance costs are to be computed, a check regarding the cost values should be made. In Figure 86, the two applications are presented as well as the services they realize. The blue-colored services refer to the policy administration functionality, while the pink-colored refer to the financial transactions. Similar overlap exists in the Infrastructure layer for the hardware related maintenance costs.

If there is no information regarding the costs of the separate functionalities, then a formula should be defined for allocating the costs to the two functionalities. To make more explicit that the same amount should not be added twice, the measure specification tables of the two measures should be refined to represent the overlap of their component measures.

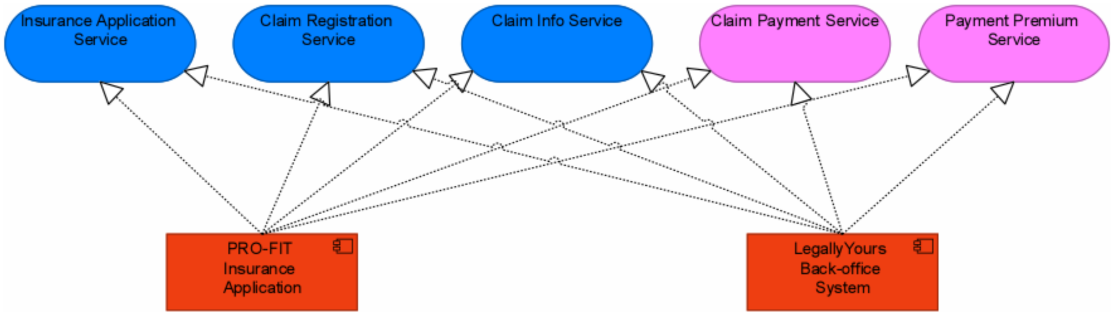


Figure 86 - PRO-FIT and Legally Yours applications overlap

Activity D2. Determine the optimal set of measures (optional)

[Assumptions] Since there are no budget restrictions in the ArchiSurance case and due to moderate ease of gaining the essential data for performing the measurements, the set of measures defined in this case is considered optimum. Thus, the application of the OMSD model is not needed.

Activity D3. Build the PI structure

Having already documented the pairs of goals and indicators as well as the measures used for calculating the lowest-level indicators, the PI Structure can now be built. In that way, ArchiSurance's goals can be directly mapped and visualized in a graph which reflects their quantitative aspect. Table 48 provides the guidelines for deriving the PI Structure relations from the goal model relations. Thus, based on Table 48, the goal-indicator pairs table (Table 67), the indicator and measure specification tables, and the target goal model itself (Figure 78), the ArchiSurance PI Structure is designed. The structure is illustrated in Figure 89. The graph was designed with the BiZZdesign Architect tool. To make the propagation of values more clear and clarify the necessary calculations for the measures (green puzzle pieces), their *component measures* have also been modeled (orange puzzle pieces) in the ArchiSurance PI Structure.

As an example, the design process of the PI structure for the goal 'Reduction of application maintenance costs' is illustrated in Figure 87. The orange puzzle pieces comprise the component measures defined in Table 70 for the 'Sum of policy administration maintenance costs' measure. The same costs are also used for measuring the 'Sum of financial applications maintenance costs' measure.

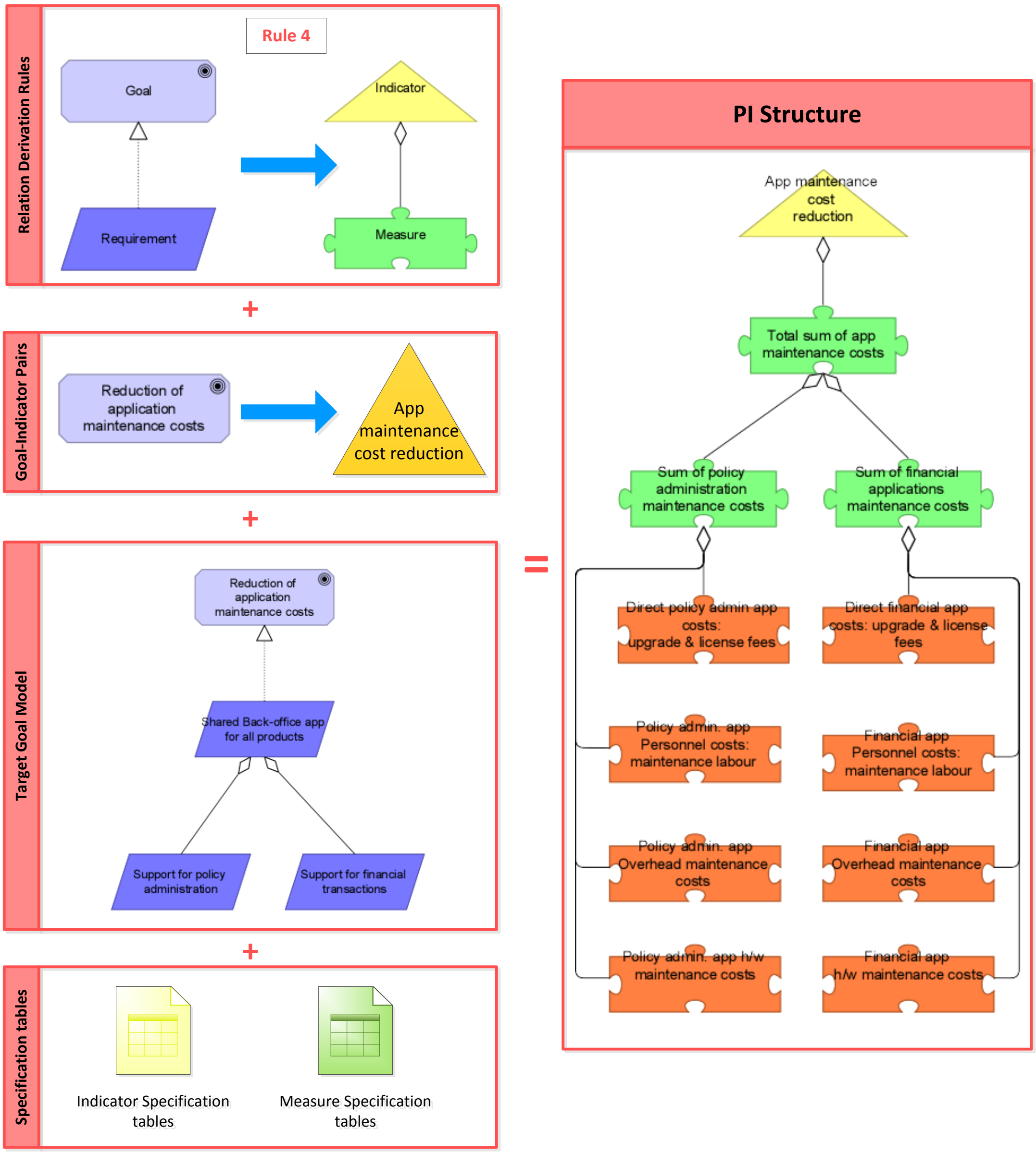


Figure 87 - Designing the PI structure for the 'Reduction of application maintenance costs' goal

Moreover, the fact that the goal 'Increase customer retention' *influences* the goal 'Increase revenue' with *no computational expressivity* between them, is reflected in the structure (Figure 88) with the *association relation* between the two indicators and the different coloring of the influencer 'Revenue retention rate increase' (blue color for the indicator). But why is the customer retention measured through revenue retention and how does revenue retention influence revenue?

To clarify the decisions on the indicators, the two terms have to be researched and defined[6]. Several blogs and sites written by experts have been consulted for deciding on the most appropriate indicators (e.g. [121] [122] [123] [124]). *Customer retention* is an indicator that measures the ability of the organization to retain its customers and to generate recurring revenue from existing customers. The amount of the recurring revenue, though, is not taken

[6] This process is part of Activity C2, where the indicators for each goal are decided. Details are provided for the goal 'Increase customer retention' in this step for clarifying the selection of the indicators and their relation in the PI structure.

into account in the calculation; which may result in misleading outcomes. Therefore, revenue retention is preferred. *Revenue retention* measures the percentage of revenue retained over a period by taking into account the number of remaining customers as well as what they have spent. Thus, even if an increase in customer retention cannot be translated in an exact revenue amount, the use of the revenue retention rate provides a less ambiguous correlation between the revenue increase and the customer retention increase. This correlation denotes that when the customer retention is increased, the revenue will most probably be increased too.

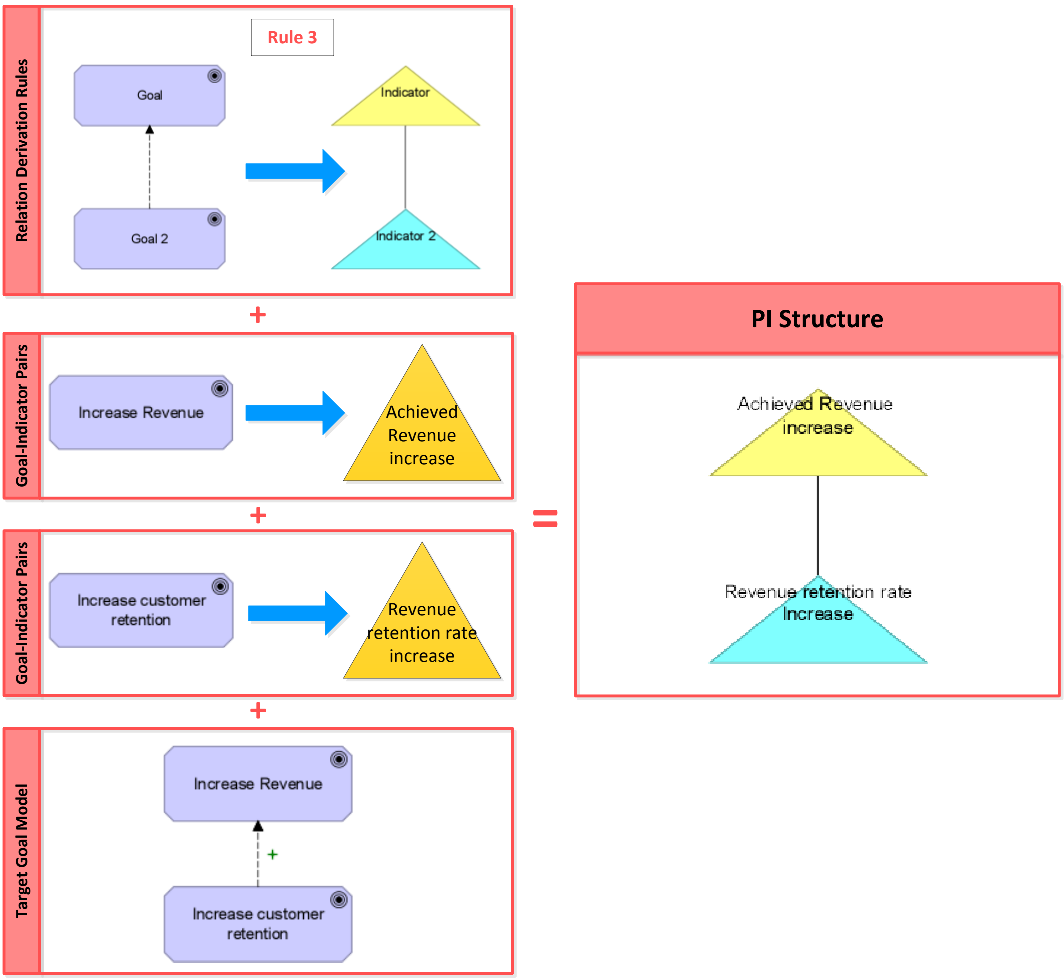


Figure 88 - Designing the PI structure for an 'influence' relation type

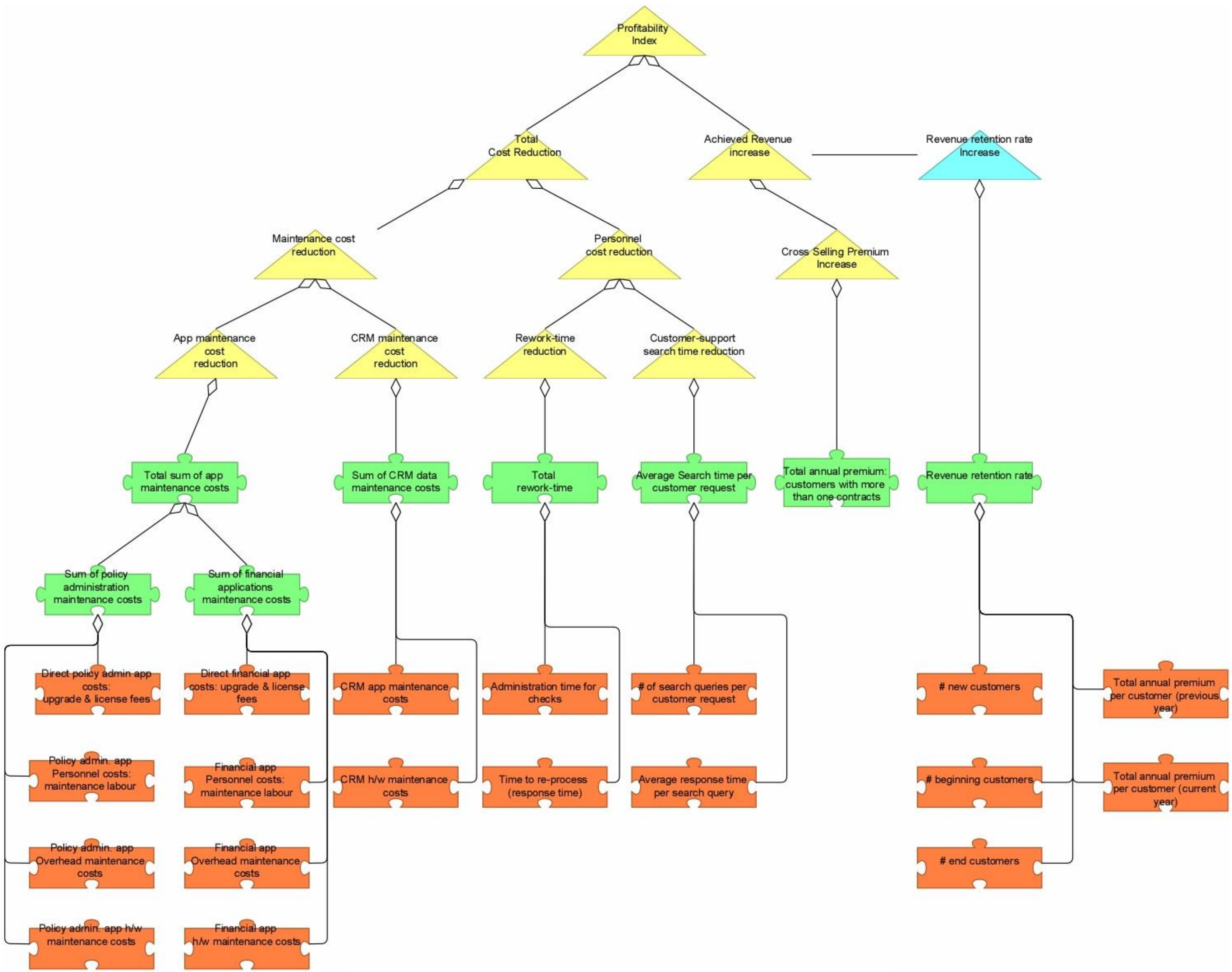


Figure 89 - ArchiSurance PI Structure

Step E: Perform the measurements in the EA model

Activity E1. Build the EA analysis model for every lowest level goal and its respective requirements

In order to isolate the measurements and perform them in a more convenient way, EA analysis models are built that facilitate in calculating the lowest level indicators. Thus, the next task that ArchiSurance needs to do for progressing in the quantitative analysis is to build the analysis models. As an example, the EA analysis model for the goal *'Reduction of data maintenance costs'* is designed in Figure 90. The analysis model includes all the components that are necessary for the calculations of the *'Sum of CRM data maintenance costs' measure*, i.e. the components that create costs related to the CRM system. By calculating this measure a first step towards calculating the *'CRM maintenance cost reduction' indicator* is made. In the baseline architecture, there are two CRM systems because of the incomplete migration during the first phase of the merger. Therefore, both systems and the concepts of the Infrastructure layer that realize them are included in the calculations.

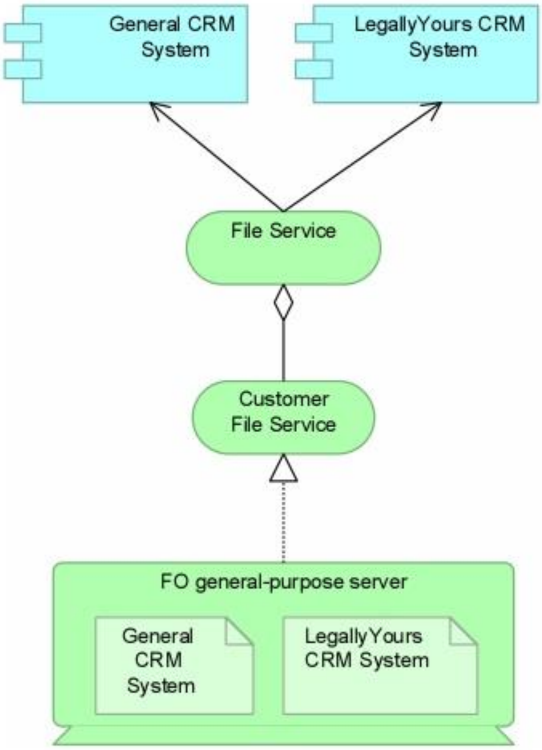


Figure 90 - 'Reduction of data maintenance costs' analysis model

While the quantitative analysis expected for the 'Sum of CRM data maintenance costs' measure is cost-oriented, another example is given to demonstrate a performance-oriented analysis as well. The focus this time is upon the 'Improve customer support' goal and its corresponding indicator 'Customer-support search time reduction'. The measure used for computing the indicator is 'Average Search time per customer request' (Figure 91), which is based on the search time that an employee needs for answering a customer request.

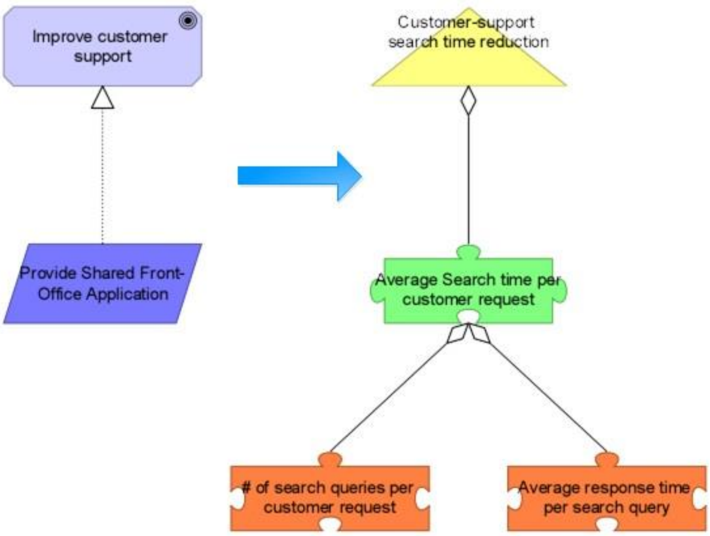


Figure 91 - 'Improve customer support' goal and its corresponding indicator and measures

The stakeholders from the ArchiSurance organization have identified as a potential source of delays in answering customer requests regarding information on their contracts or data, the lack of integrated and standardized customer data. Thus, the average *search time* per customer request needs to be calculated to provide a basis for comparing the search times in the target architecture in Step I. The search time can be viewed as the *completion time* of the process of finding the complete and correct customer information. In Table 14, completion time is defined as the *sum of response times* of the instances of the process. The response time includes the waiting time too. The 'completion time' performance measure matches the 'search time' measure, since finding the customer data may entail more than one search queries (i.e. separate tasks). Calculation formulas are proposed for indicators and measures in APPENDIX VIII. In the next activity, the analysis model for the calculation of the response time and then the completion time is designed. Additionally, the correspondence of the component measures with concepts of the EA model is also indicated.

Activity E2. Compute the measures

For the calculation of the 'Sum of CRM data maintenance costs' measure, the 'Cost Allocation across EA models' approach is going to be used (Section 3.8.3.). The component measures, i.e. the cost factors related to the CRM systems in the Application layer and the cost factors related to the servers in the Infrastructure layer are used as attribute input values for the corresponding concepts in the architecture. These individual cost factors can be found in the general ledger or the financial statement of the ArchiSurance organization. Then, a bottom-up propagation method is used for allocating and aggregating all CRM data maintenance costs to the two CRM application components. The analysis model regarding the 'Sum of CRM data maintenance costs' measure has been simplified to avoid assigning weights in the 'used-by' relations for allocating the costs from the infrastructure layer (Figure 92). Allocation based on weights would diminish the value of the calculations, since ArchiSurance is able to identify the exact maintenance costs for each CRM system.

The first analysis model would be used in case the expenses for the CRM systems were already aggregated and there was no record for the portion of the maintenance costs of each system. Then, the ABC methodology would be a convenient method for allocating costs and for calculating the exact costs of every application. An example demonstration of the ABC methodology on the ArchiSurance case is discussed in APPENDIX VIII for the cost allocation regarding policy administration application maintenance costs.

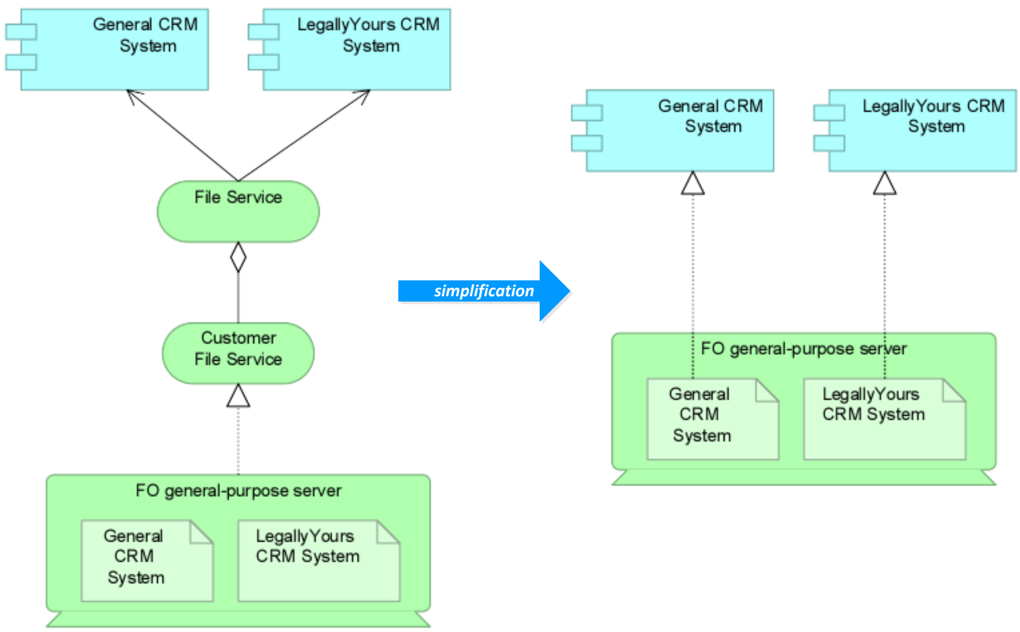


Figure 92 - Simplification of the 'Reduction of data maintenance costs' analysis model

In Figure 93, the tables next to the EA concepts provide the input values for each one of the concepts: General CRM system application, LegallyYours CRM system application, General CRM system database and LegallyYours CRM system database. Across the realization relations in the analysis model, the propagated values are also shown as a result of the calculation formulas, i.e. sum of the attribute (cost) values.

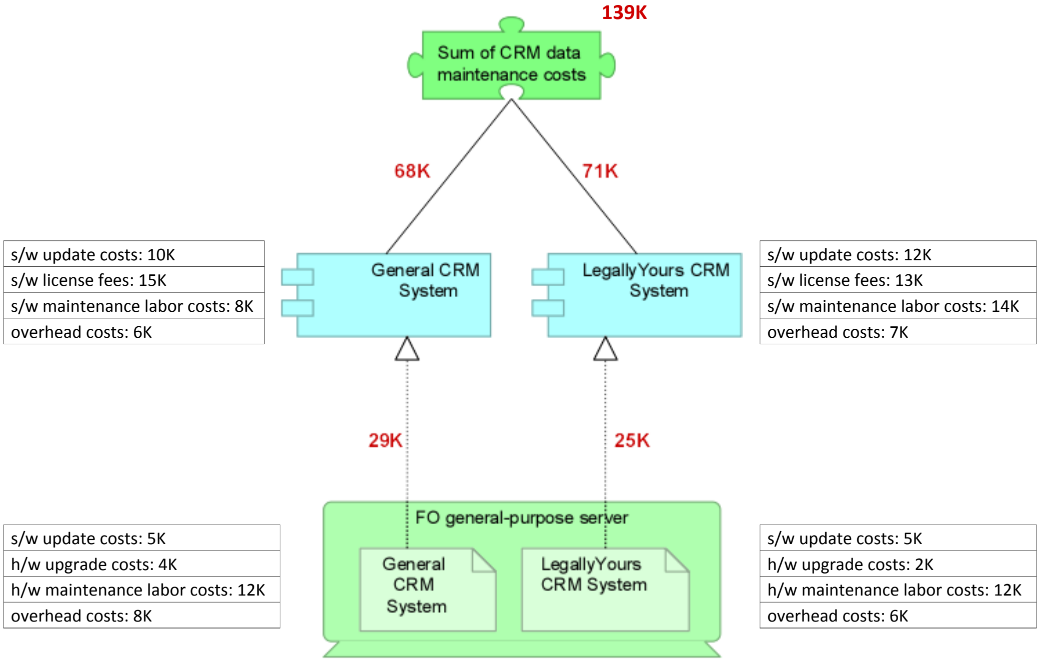


Figure 93 - Measurement of the CRM maintenance costs in Baseline EA

On the other hand, for the calculation of the 'Average Search time per customer request' measure more complicated measurement methods and functions need to be performed. For computing the search time (completion time) as the sum of response times; the 'Performance & Cost EA Analysis' approach is performed (Section 3.8.1). By using this approach, the response time for each search query will be calculated, and the utilization of the resources as well as the frequency of the requests will be also taken into account. These additional input values are presented in the pink tables in Figure 94. In general, the additional input data needed for the measurements are:

- arrival frequency of requests from the web portal *(fw)*
- arrival frequency of requests from the call center *(fcc)*
- average search time in the General CRM db per search query (service time, *Sg*)
- average search time in the LegallyYours CRM db per search query (service time, *Sly*)
- capacity of the server (*C*) and of the applications
- internal service times of the applications (*S*)

The EA analysis model comprises a simplification of the Baseline EA model. The simplified model aims to address the measurement needs. Additionally, the added 'used-by' relations between the application components bear a weight (n). It is assumed that the 2/3 of the search queries are addressed to the General CRM system which covers the Home & Away and the PRO-FIT business units, while 1/3 is addressed to the Legally Yours CRM system.

Moreover, Figure 94 shows the correspondence between the 'Average Search time per customer request' measure, its component measures and the architecture concepts. The '# of search queries per customer request' is a measure derived from the 'Customer Information service' (or the 'Inform Customer' process) and it is also used for the calculation of the response time. More specifically, the number of search queries is used as a multiplier for the arrival frequencies of the customer requests. The second component measure, i.e. the 'Average response time per search query', is the result of the measurement analysis. It is related to the 'Call Center Application', since the focus of the goal is upon the front-office.

In the analysis model, the results of the top-down workload analysis are colored red, while the response time for each element is colored green. The small tables next to the EA concepts provide all the measures that are calculated during the application of the method (T: process time, U: utilization, R: response time, λ: arrival rate or workload).

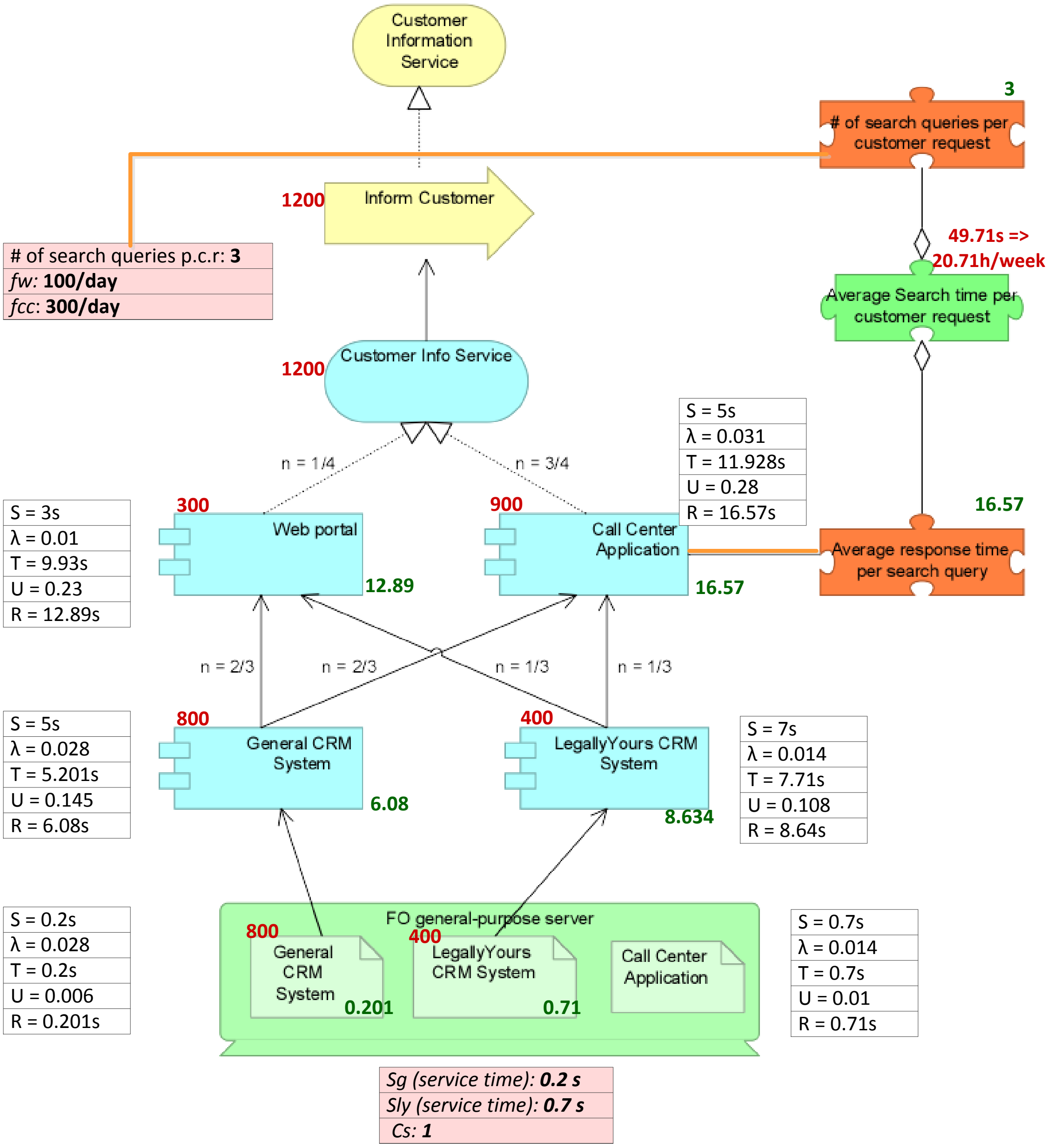


Figure 94 - Measurement of the 'Average Search time per customer request' measure in Baseline EA

Step F: Design the Target EA alternatives

Activity F1. Design the Target EA model alternatives

The requirements categorization in Step B indicated the set of requirements that are not addressed by the baseline architecture. For dealing with these issues, ArchiSurance considers two alternative target architectures. For the sake of simplicity, only one target EA alternative is going to be explained in more detail. The second is a fictitious alternative based on the ArchiSurance case, but not described in the case study [120].

Target EA alternative 1

The first alternative is designed for addressing all uncovered and partly covered requirements and for increasing the safety of the ArchiSurance organization. The changes included in this alternative are the following [120]:

1. Replacement of the separate back-office applications for the functions of policy administration and financial transactions by an integrated back-office suite.
2. Disappearance of the separate CRM system for Legally Yours
3. Replacement of the three separate general-purpose back-office servers by a shared server cluster in the data center at Home & Away headquarters, and a back-up server cluster located in the data center at PRO-FIT headquarters.

Table 72 maps the proposed changes to the requirements of Table 66 belonging in the problematic categories. Figure 95 illustrates the first Target EA model alternative.

	Partly covered	Not covered	
Changes	Provide Shared Front-office Application	Shared Back-office app for all products • Support for policy administration • Support for financial transactions	CRM data maintained centrally
Integrated back-office suite		x	
Central CRM system	x		x
Shared server cluster & back-up server cluster		x	

Table 72 - Target EA alternative 1: Changes mapped to requirements

Target EA alternative 2

The second alternative examines splitting the transformation in two phases. The first phase focuses on addressing the goal of 'Reduction of application maintenance costs' and, thus, migrating to a shared back-office application suite. In this alternative the existing complexity in the application landscape regarding the main services of the organization is prioritized. ArchiSurance is planning to integrate the CRM systems after the application landscape is simplified. By doing so, the organization expects less costs and a shorter transformation period. Not integrating the CRM systems, though, results in not covering the requirements related to data standardization. This EA alternative model is not further worked out, but it is going to be used for supporting the comparisons in Step J.

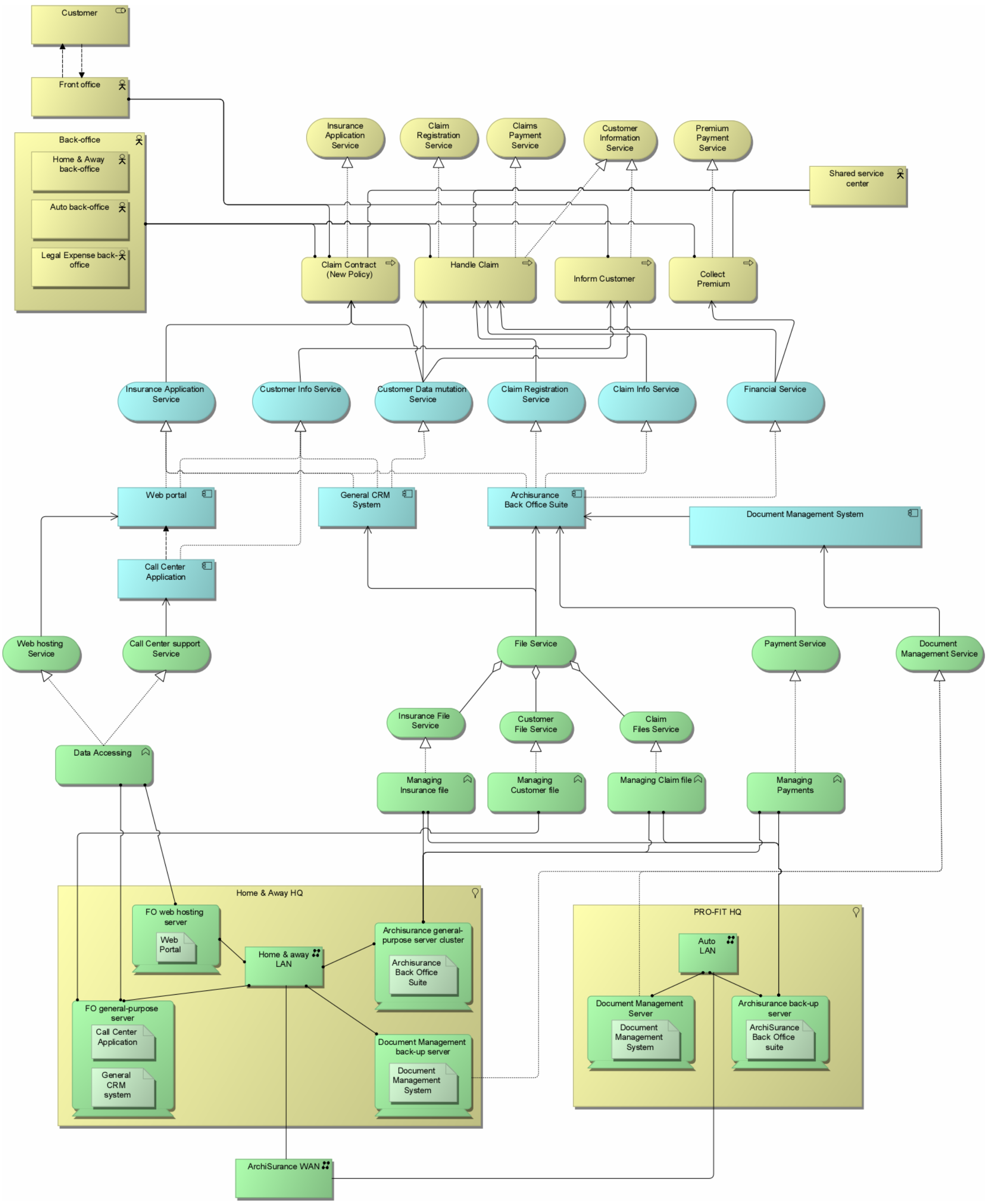


Figure 95 - Target EA alternative 1

Activity F2. Relate the Target EA concepts with the Requirements

Since the Target EA alternative 1's proposed changes address the requirements' needs, the relation type to be used for mapping the requirements to target EA concepts is the realization relation. A part of the requirements' realization viewpoint in the target architecture is illustrated in Figure 96. Figure 96 demonstrates the realization viewpoint for the group of requirements which realize the goal 'Reduction of application maintenance costs'.

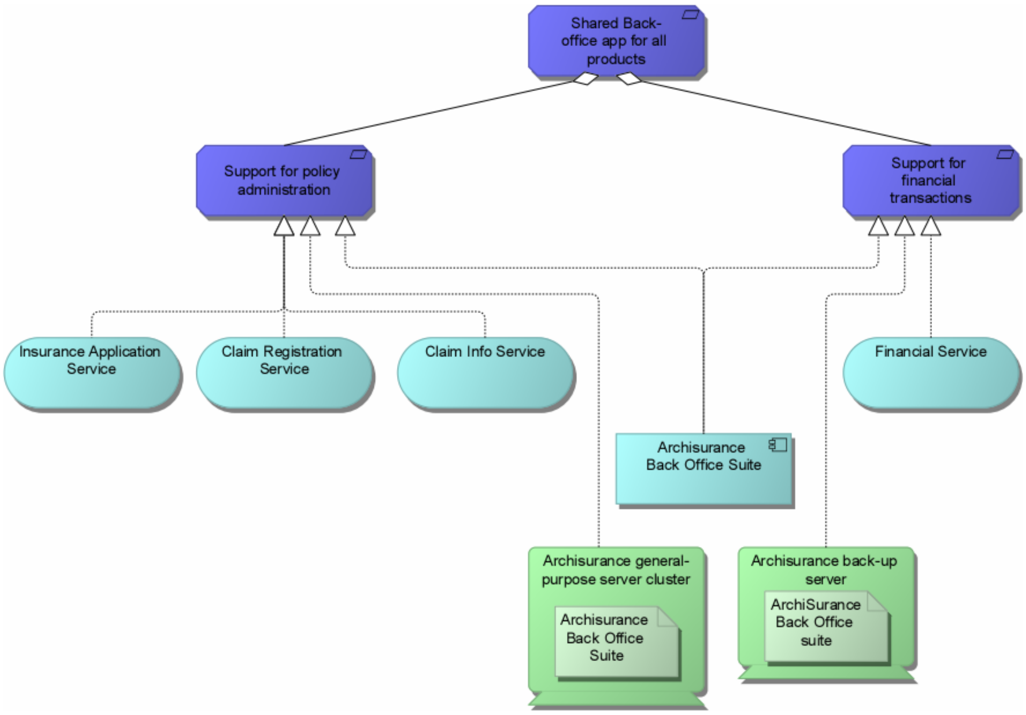


Figure 96 - 'Shared back-office app for all products' requirement realization in Target EA model

Step G: Define the measures for the Target EA alternatives

Activity G1. Redefine the measures of the indicators for the Target EA alternatives

The 'type of measure' dimension of the measures specified for the baseline architecture remains the same, but the 'type of monitoring objects and points' changes for the target EA model. For assigning the new monitoring objects to the measures, the analysis models can be designed in parallel. The analysis models for the goals 'Reduction of data maintenance costs' and 'Improve customer support' (same as in Step E) are provided in Step H.

To indicate the changes in the measure specifications in this activity, the analysis model for the goal 'Reduction of application maintenance costs' is chosen. This model is valuable to demonstrate because the target EA model causes changes not only in the realization of the requirements but in the component measures for quantifying the application maintenance costs and consequently the PI structure as well.

In the target EA model, only one integrated application exists for realizing the policy administration and the financial transactions comparing to the four different applications in Baseline EA (Figure 97).

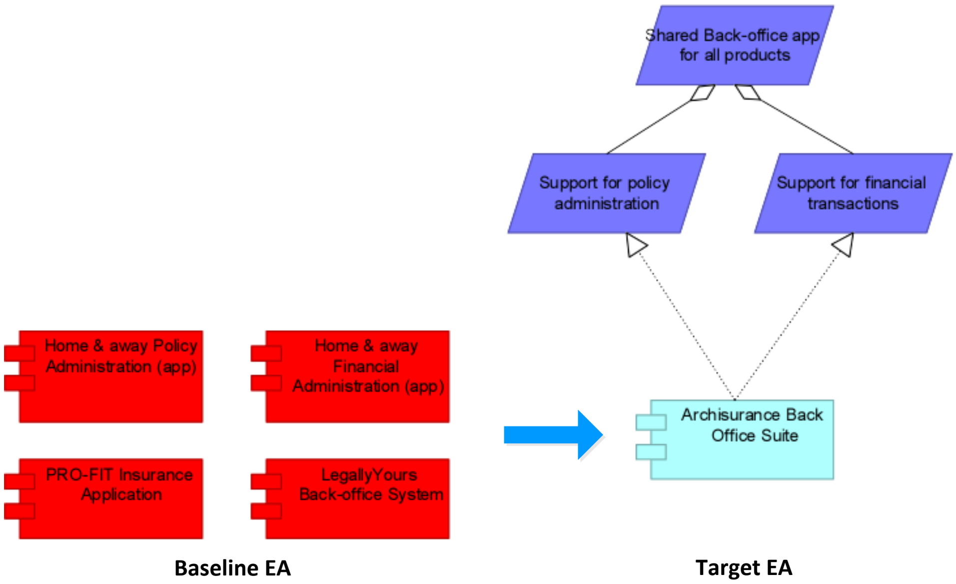


Figure 97 - ArchiSurance target shared Back-Office application

Since the 'ArchiSurance Back-office suite' realizes both requirements (Figure 97), the measure 'Total sum of app maintenance costs' which corresponds to the 'Shared Back-office app for all products' requirement will be directly derived from the aggregated costs at the application component (Figure 99).

Furthermore, the changes in the Infrastructure domain impose changes in the costs derived from this domain as well. The decision of ArchiSurance to have a shared server cluster and a back-up server cluster in different locations indicates that the maintenance of the network is also important for achieving the mirrored data storage (Figure 98).

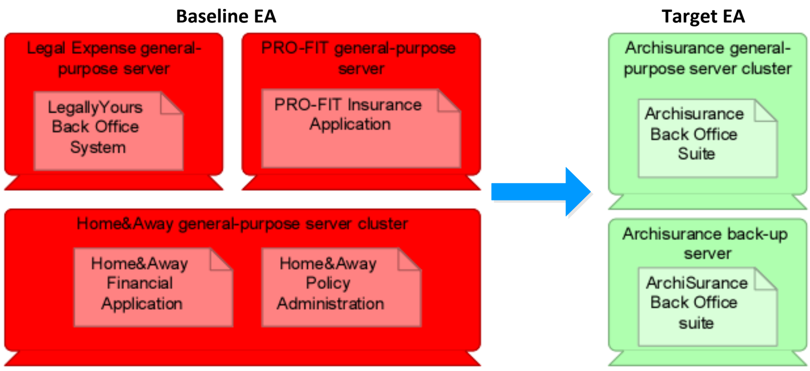


Figure 98 - ArchiSurance target shared server cluster

The analysis model for the measurement of the application maintenance costs and the changed corresponding measures of the PI Structure are presented in Figure 99.

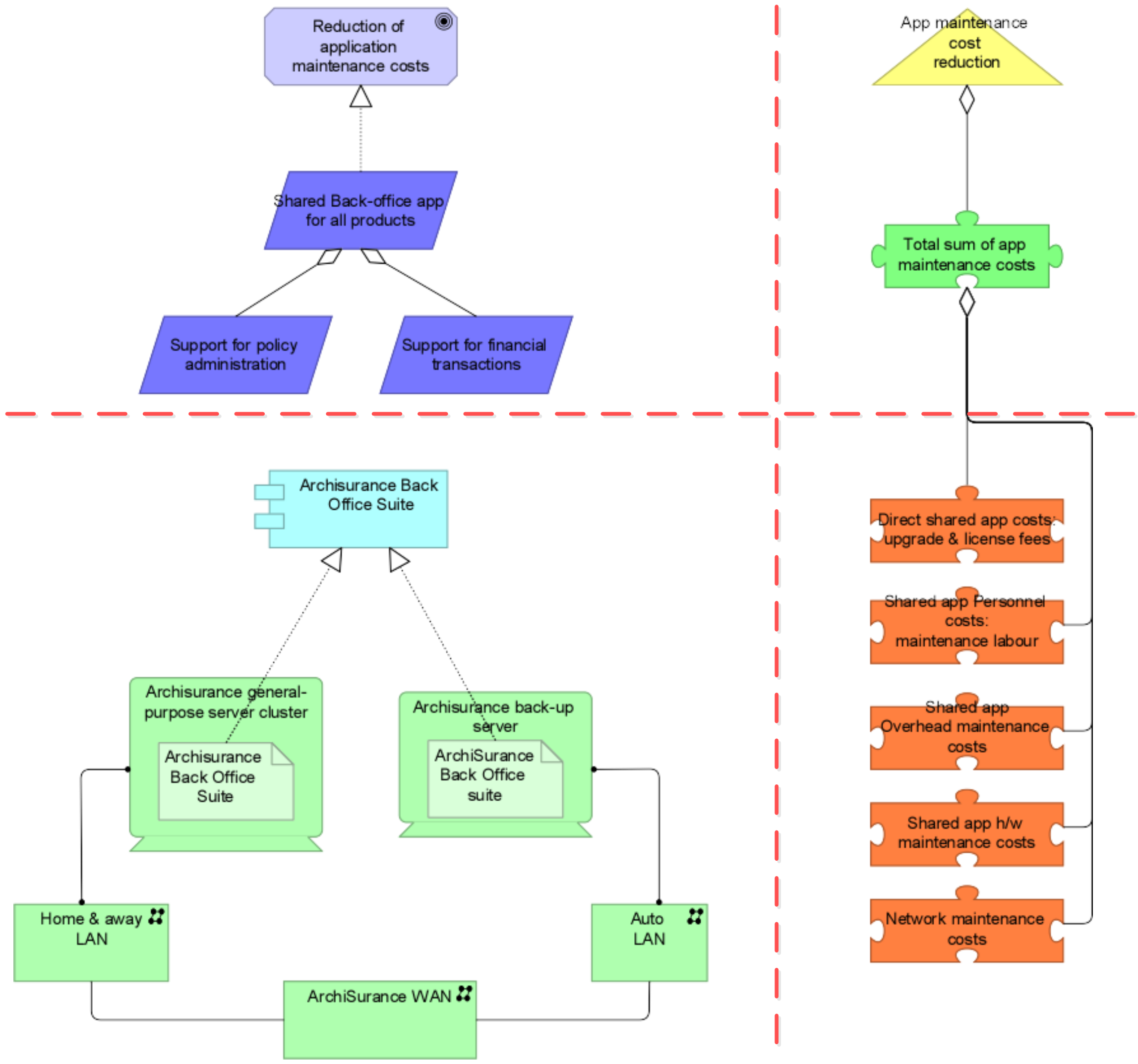


Figure 99 - Analysis model and PI structure for measuring the Shared app maintenance costs in Target EA

Activity G2. Determine the focus area of each goal in the Target architecture alternatives

By building the analysis models for the requirements and consequently for the lowest-level goals, the focus area can be determined by rolling back the goal model. This activity is skipped for the target architecture in this demonstration, since it has been already discussed for the baseline architecture.

Step H: Repeat Step E for all Target EA alternatives

Activity H1. Build the EA analysis model for every lowest level goal and its respective requirements

The analysis models for the goals 'Reduction of data maintenance costs' and 'Improve customer support' (same as in Step E) are presented in Figure 100 and Figure 101 respectively.

Activity H2. Compute the measures

The same methodologies, as in Activity E2, are used in this activity for measuring the maintenance costs related to the CRM system and the search time per customer request. The input values for the target architecture, though, are just estimated or expected values.

Figure 100 presents the measurement and the propagation of the CRM application maintenance costs. The tables in Figure 100 contain the expected cost input values for the calculations of the maintenance cost measures in the Target EA alternative 1. The only difference from the calculations performed in the Baseline EA refers to the time aspect. Since the amounts of money measured will occur in the future, the *time value of money* needs to be considered and the costs must be brought to the present value of money through the use of the discount rate (Dynamic measures, Chapter 3). The timeframe of the architecture solution is used for denoting the future point in time. The formula for calculating the present value of money is the following:

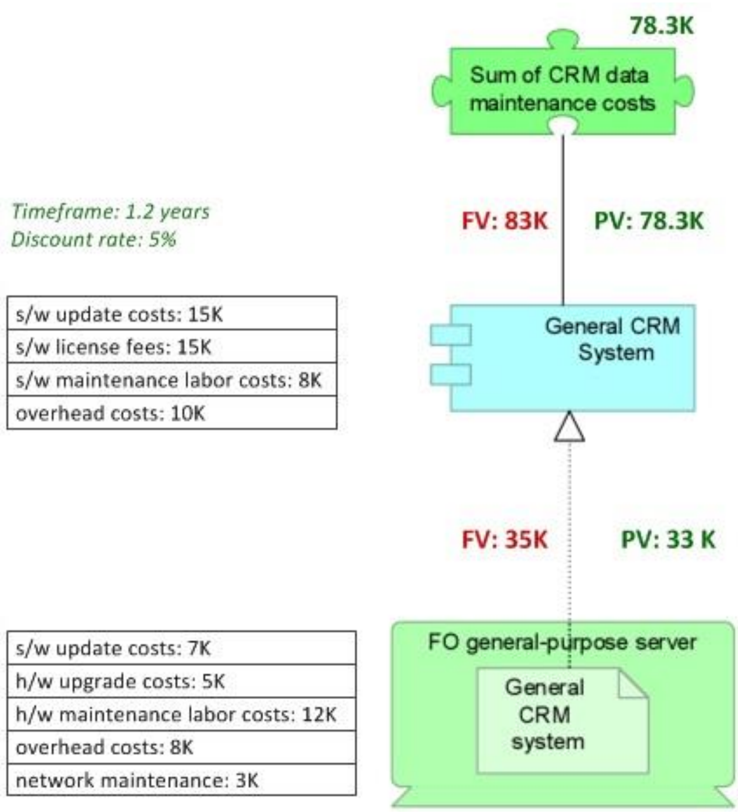


Figure 100 - Measurement of the CRM maintenance costs in Target EA

$$PV = \frac{FV_t}{(1+r)^t}$$

As for the calculation of the 'Average Search time per customer request' measure, similarly to the analysis method carried out in Activity E2, Figure 101 presents the input and output values as a result of the top-down and bottom-up propagations. The arrival frequencies for the web portal (*fw*) and the call center (*fcc*) have remained the same.

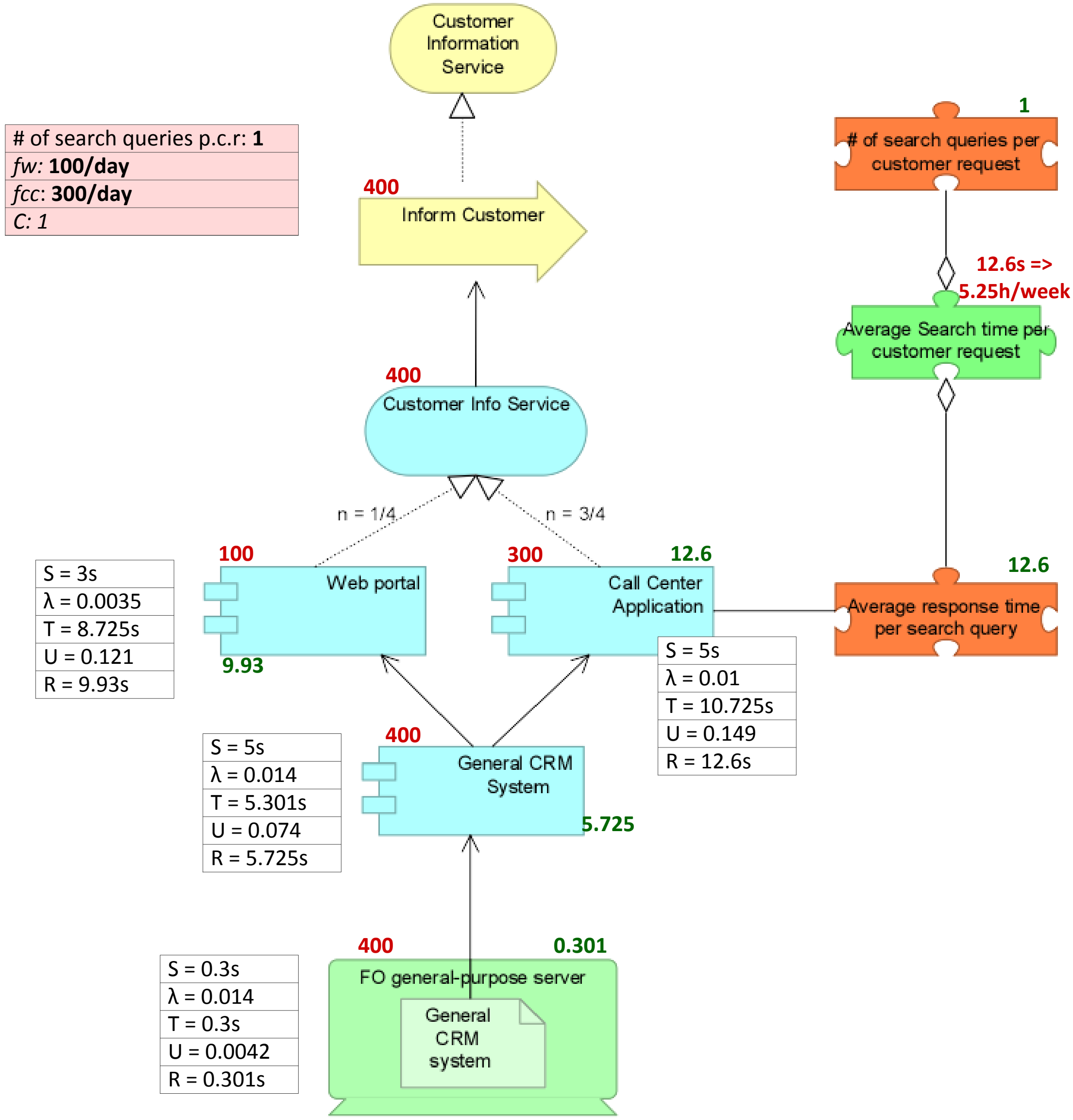
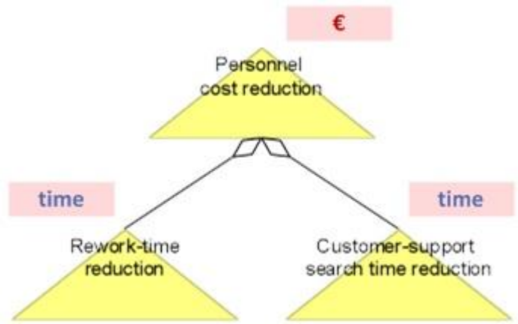


Figure 101 - Measurement of the 'Average Search time per customer request' measure in Target EA

Step I: Perform Indicator Analysis for every Target EA alternative

Activity I1. Calculate indicators based on the PI structure

After the measurements have been completed, the indicators are calculated based on the calculation formulas proposed for each indicator in APPENDIX VIII. The computed measures from the Steps E and F are the input values for the indicators in the bottom-level of the PI structure, where the propagation of the values starts. The output values, i.e. the current values, of the indicators are presented in Figure 104 for Target EA alternative 1.



From the calculation formulas used in the ArchiSurance case, one needs to be further explained. That is the formula of the 'Personnel cost reduction' indicator. The scale of this indicator is in €, while the scales of the indicators below it are in employee-time. To translate the employee-time into costs, the 'Time Savings Time Salary' model is used. This model assumes that by introducing a new technology, time can be saved which can be invested elsewhere. Based on this assumption, the cost reduction offered by the changes in the architecture is calculated by utilizing the average wage (W) of the employees involved in the processes analyzed and the total time (T) an employee works per week. These variables can be also viewed as conversion factors (BIP approach) for expressing the computational relation among the three indicators.

Furthermore, in this activity the timeframe of each indicator can be determined depending on the alternative that is analyzed. In the Indicator Specification Table, the timeframe is also used as an influencer of the performance level of the indicator based on the following formula. For example, for the 'App maintenance cost reduction' indicator (Table 69), if the time frame is 1.8 years, the time-difference multiplier would be:

$$time_{difference}(td) = \frac{t_i - t_g}{t_g} = \frac{1.8 - 2}{2} = -0.1 \Rightarrow final\ pl > pl$$

where, t_i is the timeframe of the indicator = 1.8 years

t_g is the timeframe of the goal = 2 years

The time-difference values for each indicator are also presented in Figure 104 as *td*. An example of the value propagation across the PI structure is provided in Figure 102.

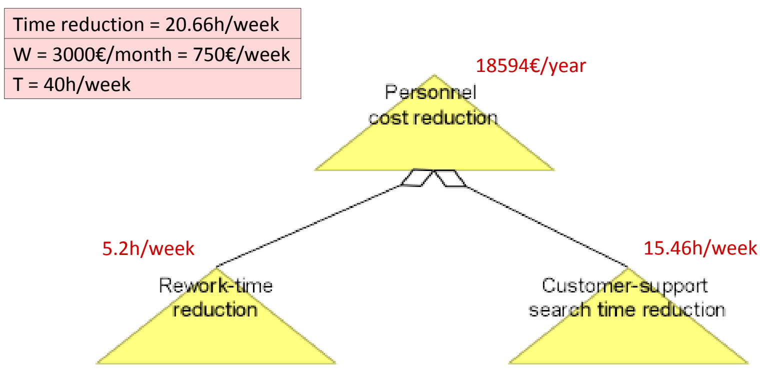


Figure 102 - 'Personnel cost reduction' indicator: Current value calculation

Activity I2. Calculate indicator performance levels

The indicator specification tables are used in this activity for normalizing and interpreting the indicators' performance. Based on the target, threshold and worst values specified for each indicator by the stakeholders in the ArchiSurance organization, the performance levels and the performance regions the indicators fall into are computed. The performance levels and performance regions of all indicators in the ArchiSurance case are presented in Figure 104. The time-difference multipliers have been also used for the calculation of the final performance value of each indicator.

Figure 103 depicts the performance levels and performance regions for the 'Personnel cost reduction' indicator and its lower-level indicators. As it can be observed, the 'Customer-support search time reduction' indicator performs ideally, while the 'rework-time reduction' indicator shows very little improvement. The low performance of this indicator was expected due to the fact that no significant changes have occurred in the architecture to directly affect the realization of the corresponding goal, i.e. the 'Reduction of rework'. The contradiction between the performance levels of these two indicators is also obvious at the performance level of the 'Personnel cost reduction' indicator, which is moderate. The colored circles in Figure 103 denote the performance regions. The change in the performance levels and regions by taking into consideration the time-difference multipliers is also illustrated. Since the timeframe of the CRM integration is 1.2 years comparing to the 1 year of the corresponding goal, the performance levels are lower but only the 'Customer-support search time reduction' indicator's performance region changes from 'best' to 'very good'.

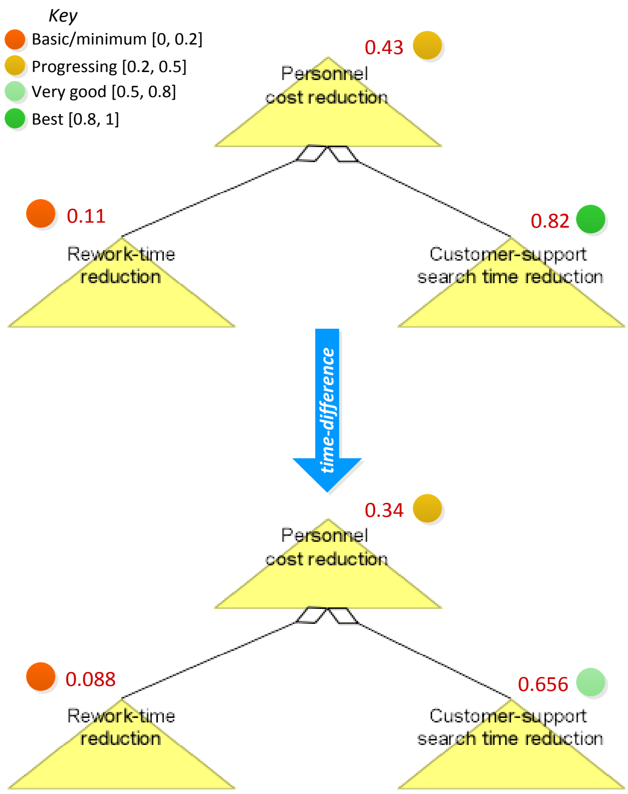


Figure 103 - 'Personnel cost reduction' indicator: Performance level & region

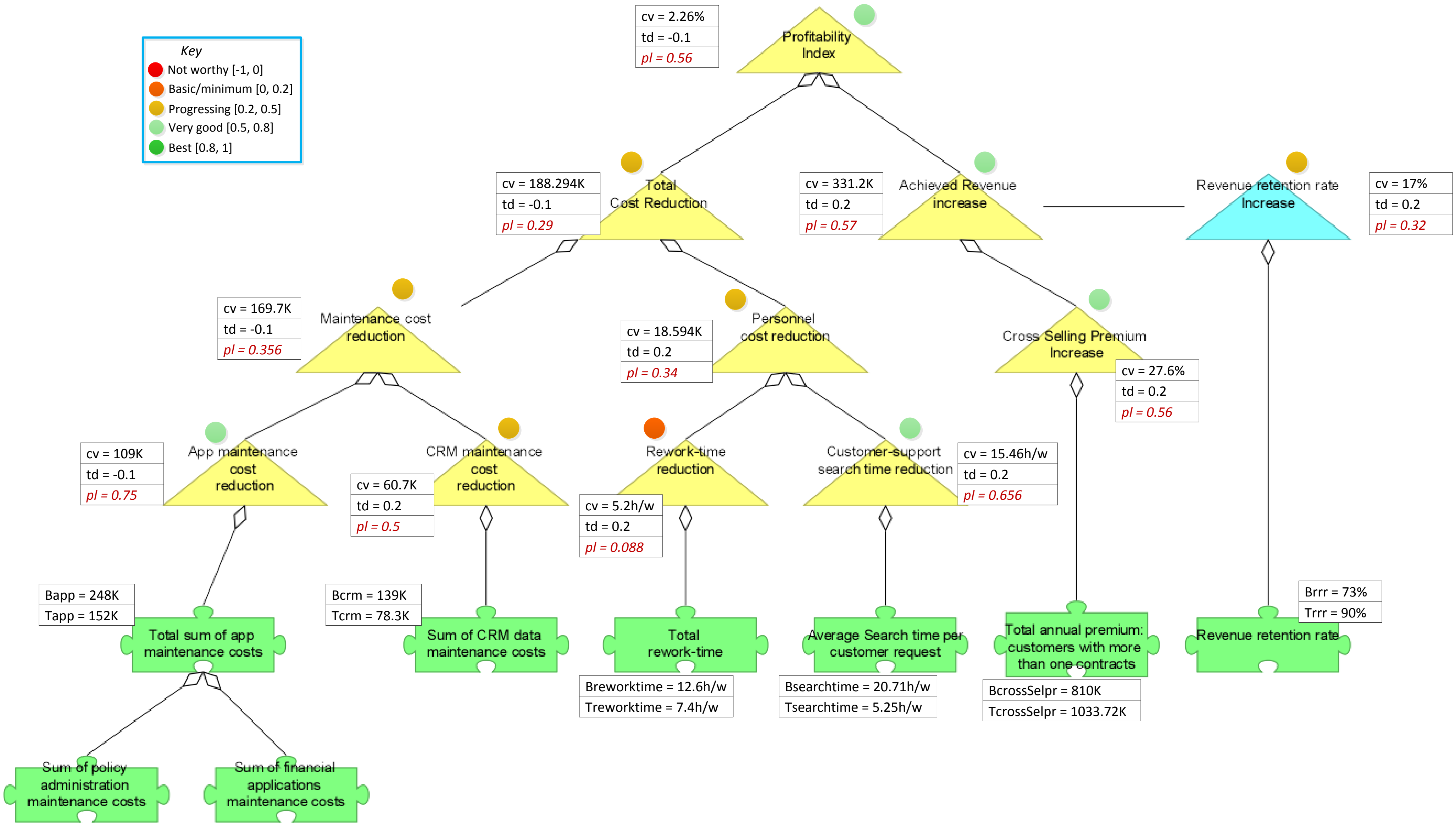


Figure 104 - ArchiSurance PI Structure: Current values, Performance levels and Performance regions

Activity I3. Map indicator performance levels to goal satisfaction levels

The interpretation of the performance level of each indicator and goal can be reflected either on the PI structure or on the target goal model.

Figure 105 depicts the goals being measured through the 'Personnel cost reduction', 'Customer-support search time reduction' and 'Rework-time reduction' indicators. The performance levels computed in Activity I2 have been assigned as satisfaction values to goals, while the corresponding performance range is represented by coloring the goals accordingly.

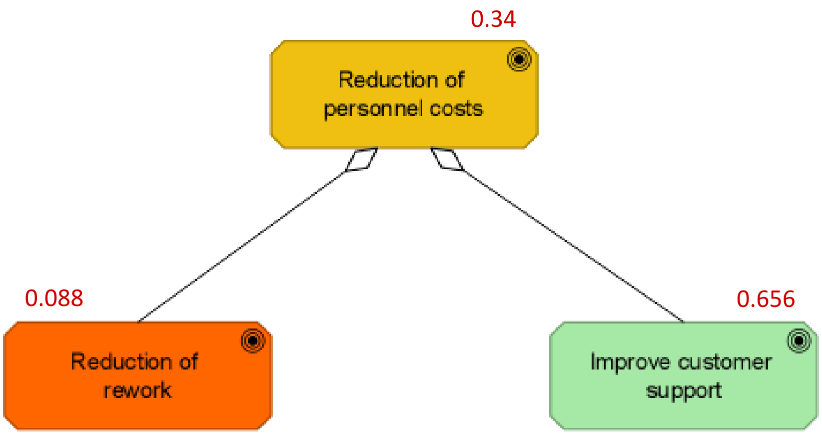


Figure 105 – Goal satisfaction values for the 'Reduction of personnel costs' goal

In Figure 106, the satisfaction values of all goals in the ArchiSurance goal model are provided for Target EA alternative 1, while Figure 107 presents the colored goal model with the corresponding satisfaction values for Target EA alternative 2, where the underperformance of the indicators depending on the integration of the CRM systems is obvious.

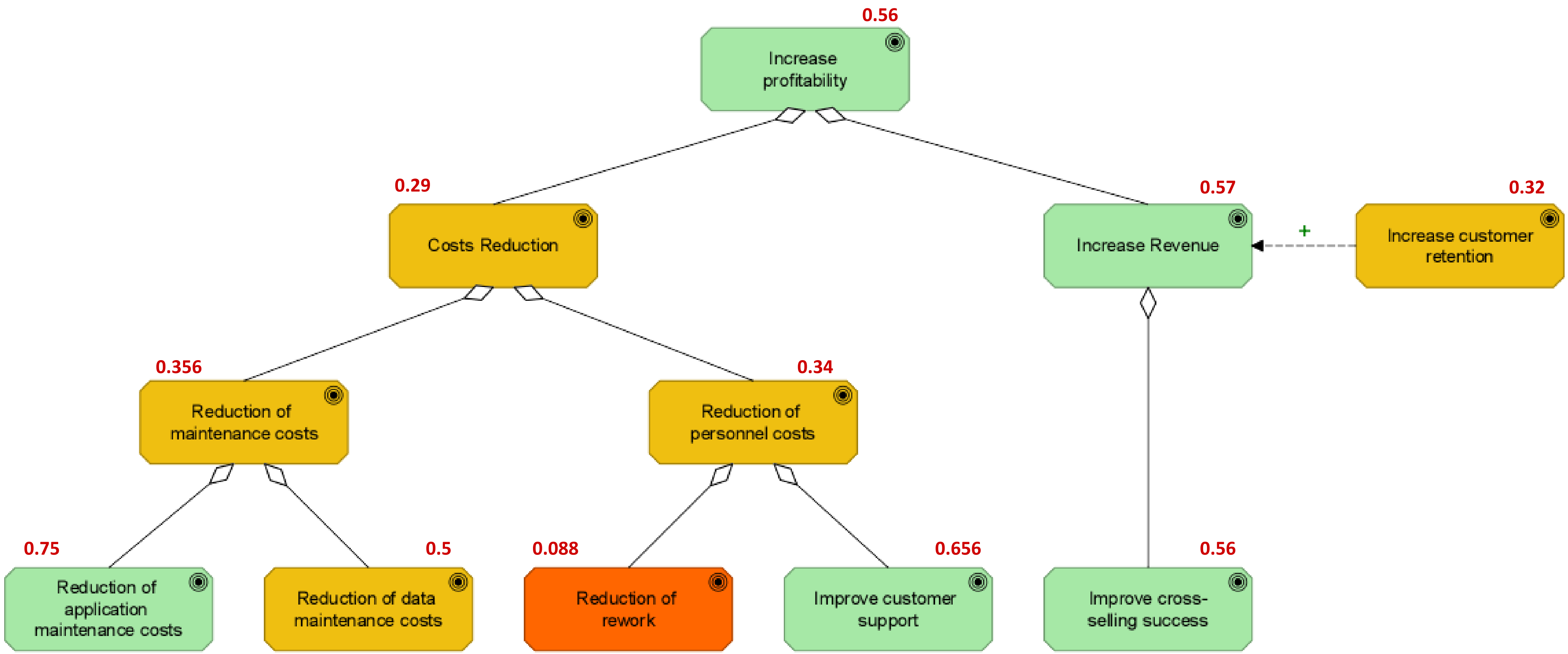


Figure 106 - ArchiSurance Goal Model: Goal Satisfaction Values (Target EA alternative 1)

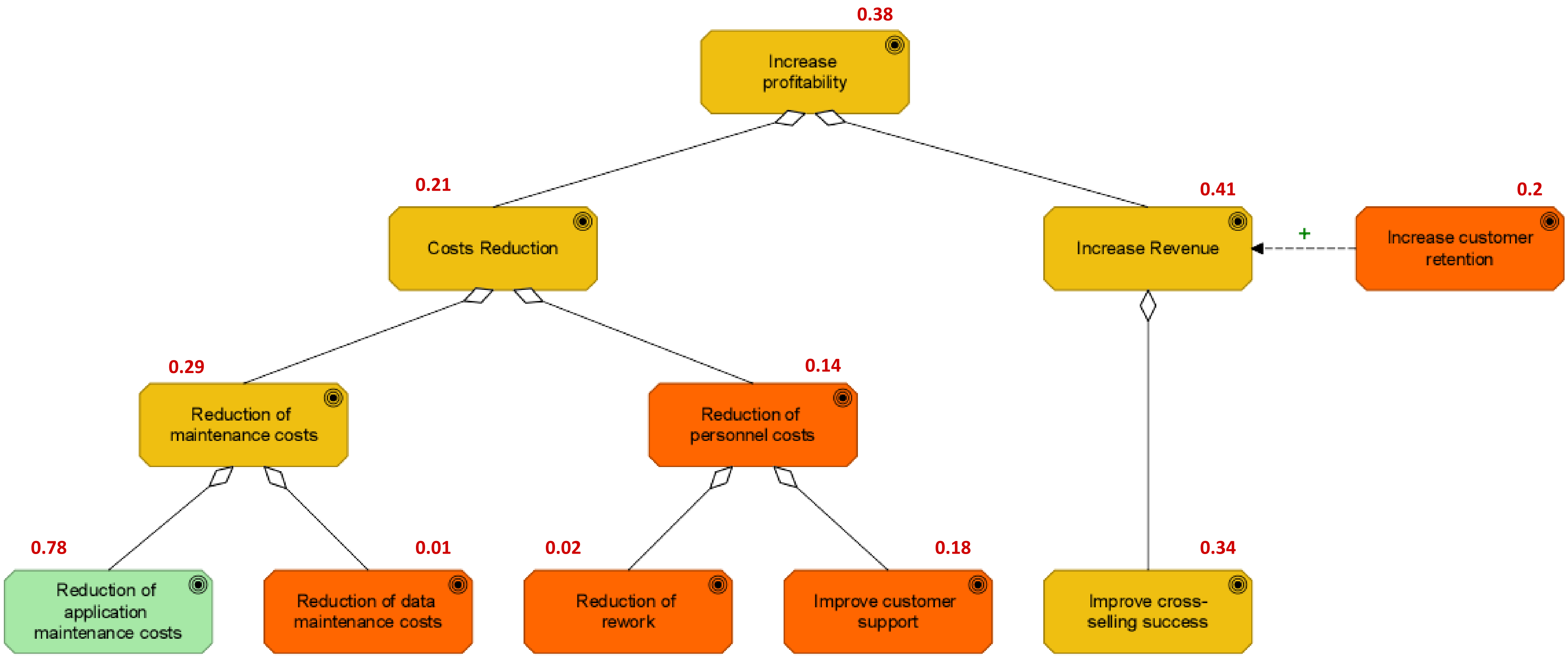


Figure 107 - ArchiSurance Goal Model: Goal Satisfaction Values (Target EA alternative 2)

Step J: Compare alternatives – Overall Analysis

Activity J1. Analyze the goal satisfaction values and make comparisons for all alternatives

The analysis of the ArchiSurance goals has been completed with respect to the Baseline EA and the Target EAs. The organization now has to decide which alternative to choose by evaluating the outcome of the analysis. The evaluation will address the overall satisfaction of goals and the complexity of the architectures.

By observing the colored goal models for the two alternatives (Figure 106 and Figure 107), ArchiSurance can realize that both target architectures offer advantages comparing to the baseline model, since all satisfaction values are positive and no goal is colored red. Additionally, the range of coloring denotes that the target EAs perform above the threshold values determined in Step C. A one-to-one comparison of the goal satisfaction values can be represented in a column chart for making clearer the performance differences between the two solutions. The visualization of this comparison is part of Activity J2.

One of the main drivers for performing the analysis and defining the target goals was the complexity of the baseline architecture due to the existence of overlapping applications and consequently databases. By observing the focus area of the goal 'Reduction of maintenance costs', which includes all the applications and infrastructure elements related to policy administration, financial transactions as well as CRM data, the complexity between the baseline EA and the target EA1 can be compared (Figure 108). By just counting the elements in each layer, a 39% reduction is achieved in the Application layer and a 67% reduction in the Infrastructure layer (based on the number of databases). Additionally, the installation of the back-up server provides more security to the organization.

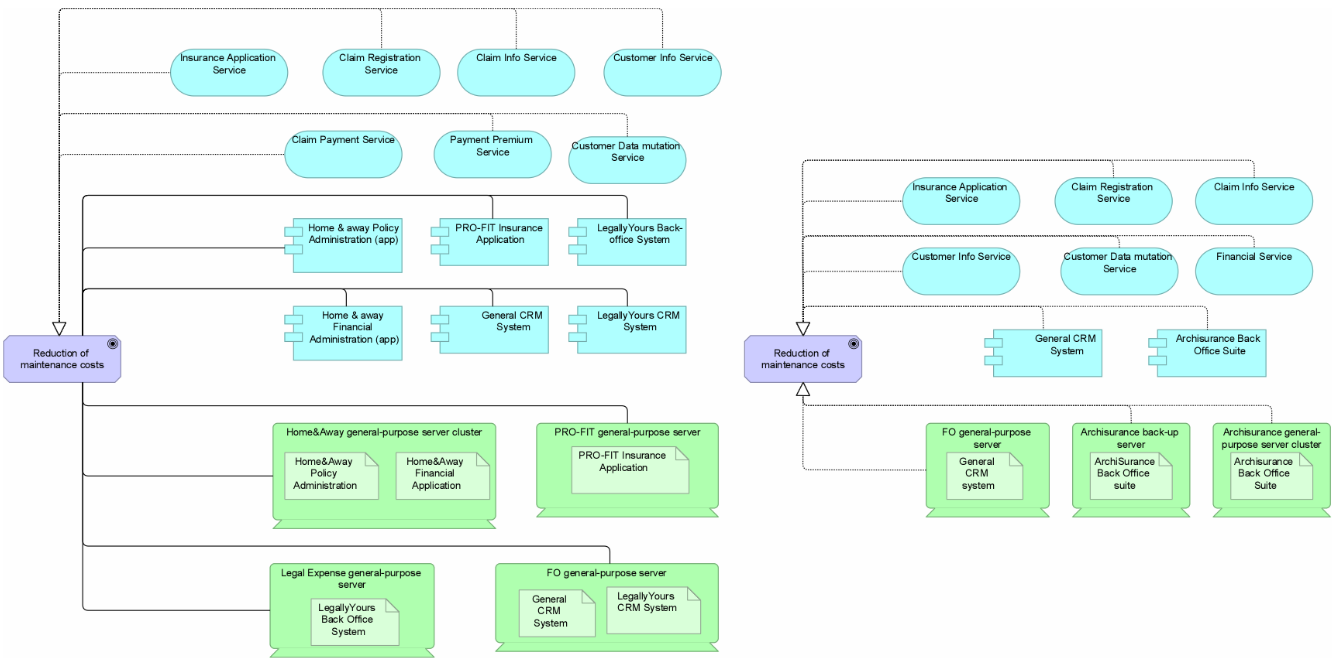


Figure 108 - Complexity reduction: Focus area comparison for the 'Reduction of maintenance costs' goal

Activity J2. Visualize the analysis/comparison results

The visualizations proposed in Activity J1 as well as the visualizations of the indicators proposed in Activity C3 are designed in order to support the decision-making and to clarify the differences among the alternatives.

The column chart in Figure 109 presents the comparison of the goal satisfaction values for the two EA alternatives, where the overall higher performance of target EA1 is apparent. The pie chart in Figure 110 represents the portion of the total cost reduction in target EA alternative 1 that the maintenance cost reduction and personnel cost reduction cover. Finally, the bar chart in Figure 111 demonstrates the absolute values of the 'App maintenance cost reduction' indicator for the two alternatives, the expected timeframe for each one as well as their performance level. This chart can explain why there is a difference in the performance levels of the indicator for the two alternatives; which is actually the timeframe of the implementation. The small timeframe difference as well as the small performance level difference indicate that the second alternative solution does not offer a substantial improvement even for this focus area.

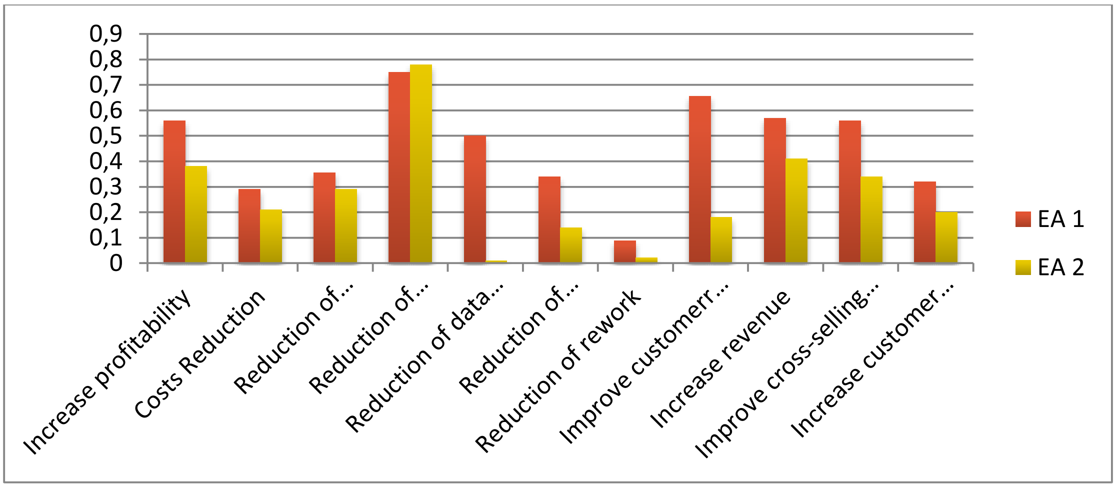


Figure 109 - Goal satisfaction values comparison for target EA alternatives: Column chart

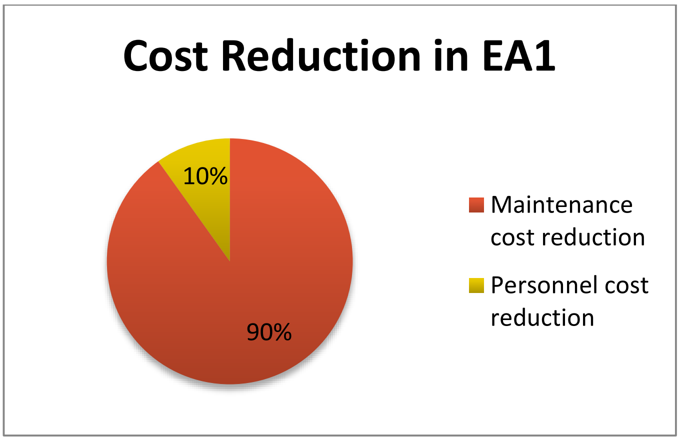


Figure 110 - Total cost reduction in EA1: Pie chart

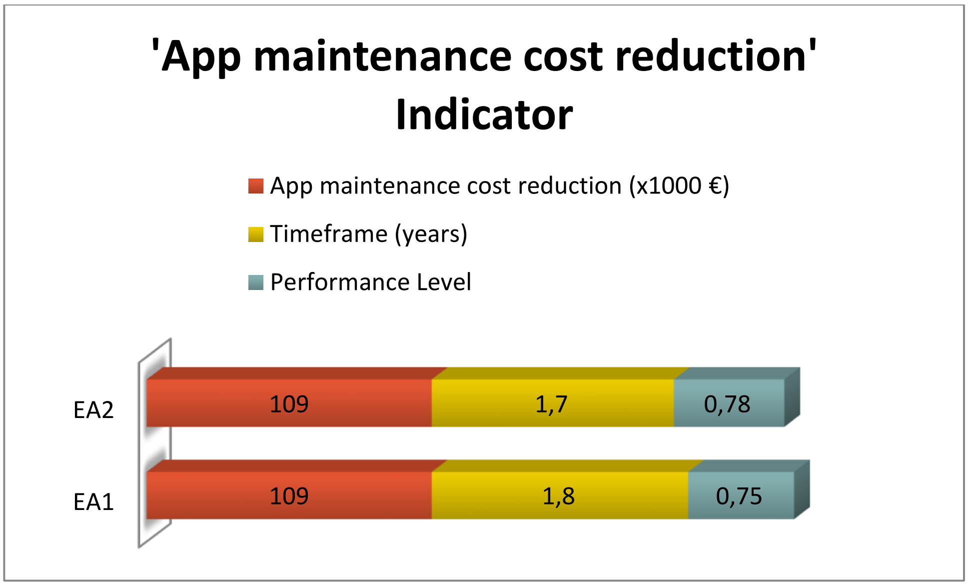


Figure 111 - 'App maintenance cost reduction' indicator comparison: Stacked Bar chart

Activity J3. Incorporate quantitative goal change analysis with business case analysis techniques

The last activity comprises the incorporation of additional information in the analysis of the two alternatives. Figure 112 compares target EA1 and EA2 with respect to the migration costs involved for realizing each solution and the timeframe for realizing the benefits. The bubbles represent the cost reduction and revenue increase absolute values as they were calculated for each alternative in Step I. So, even if the migration costs for EA1 are higher, the expected benefits are larger as well, which is denoted by the bigger bubbles representing EA1's benefits. Therefore, ArchiSurance is going to implement EA1 since the expected benefits are higher and the overall goal performance is better.

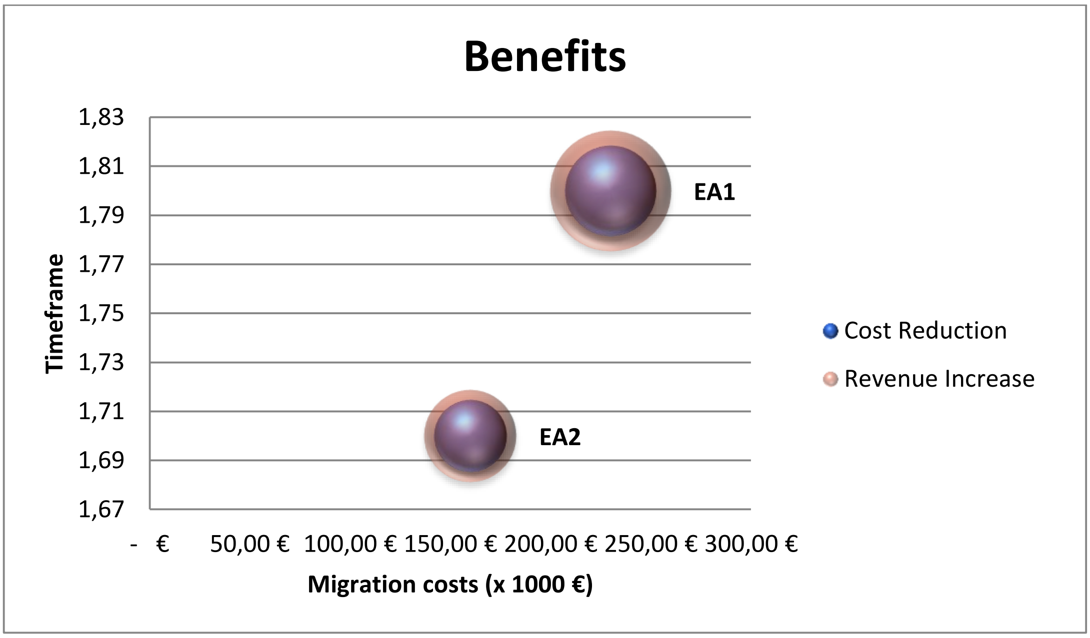


Figure 112 - Benefits comparison between target EAs: Bubble chart

6.3. Implementing the indicator value propagation – Algorithm

To demonstrate that the EAGQ method can be implemented in a tool (such as Architect), a set of algorithms are provided in this section. The algorithms are in fact pseudocode and they cover only the Activities I1 and I2 of the EAGQ method, where the propagation of the absolute values across the PI structure is performed and the calculation of the performance levels is realized. The attributes specified for each indicator and measure in Activity C2 are treated as variables in the algorithms below, except from the 'calculation formula' which is treated as a function. These formulas operationalize the computation of the value of each indicator.

In the algorithm below, the absolute values of the indicators are propagated across the PI structure starting from the bottom level of the structure. For every calculated value, its corresponding performance level is also computed with the use of the target value, threshold value and worst value.

Algorithm for propagating values across the PI structure

```
level_i = 0;
max_level = 0;
for each goal
{
        If (goal.level > max_level)
        {
                max_level = goal.level;     //find the maximum goal level in the goal model
        }
}
while (level_i =< max_level)          //repeat for every level of goals, i.e. indicators
{
        for each indicator.goal.level == level_i        // find the children of each indicator
        {
                for each indicator.child     // find the value of each child
                {
                        if (indicator.child.type == 'measure')
                        {
                                // get Baseline and Target value
                                measValB[] = indicator.child.bvalue;
                                measValT[] = indicator.child.tvalue;

                                //get the measure's name to distinguish the tables' values
                                measNames[] = indicator.child.name;

                                c = 0;     //variable that shows that children are measures
                        }
                        else if (indicator.child.type == 'indicator')
                        {
                                indVal[] = indicator.child.value;

                                //get the indicator-child's name to distinguish the table's values
                                indNames[] = indicator.child.name;

                                c = 1;     //variable that shows that children are indicators
                        }
                }
```

```
                    if (c==0)           //calculate the value of the indicator from its measures
                    {
                            indicator.value = indicator.calculation_formula(measValB[], measValT[],
measNames[]);
                            empty measValB[];
                            empty measValT[];
                            empty measNames[];
                    }
                    else if (c==1)   //calculate the value of the indicator from its indicators
                    {
                            indicator.value = indicator.calculation_formula(indVal[], indNames[]);
                            empty indVal[];
                            empty indNames[];
                    }

                    //calculate the performance level of the indicator
                    if (indicator.value >= indicator.threshold)
                    {
                            indicator.performance_level = (indicator.value - indicator.threshold)/
                            (indicator.target - indicator.threshold);
                    }
                    else if (indicator.value < indicator.threshold)
                    {
                            indicator.performance_level = (indicator.value - indicator.threshold)/
                            (indicator.threshold - indicator.worst);
                    }
            }
            level_i++;          //move to the next (higher) level
            c = -1;             //initiate the variable
}
```

The following two algorithms comprise examples of functions for the calculation formulas of the indicators ‘Total cost reduction’, which is calculated by two other indicators, and ‘Customer-support search time reduction’, which is calculated by the baseline and target values of a measure.

Calculation formula example for the indicator ‘Total cost reduction’

```
double calculation_formula(double values[], string names[])
{
            double CRtotal;
            double CRmaintenance, CRpersonnel;

            //find which value correspond to which indicator
            if (names[0] == ‘Maintenance cost reduction’)
            {
                    CRmaintenance = values[0];
                    CRpersonnel = values[1];
            }
            else
            {
                    CRmaintenance = values[1];
                    CRpersonnel = values[0];
            }

            CRtotal = CRmaintenance + CRpersonnel;
```

```
        return CRapp;
}
```

Calculation formula example for the indicator 'Customer-support search time reduction'

```
double calculation_formula(double Bvalues[], double Tvalues[], string names[])
{
        double SupportTime, B_SearchTime, T_SearchTime;

        if (names[0] == 'Average Search time per customer request')
        {
                B_SearchTime = Bvalues[0];
                T_SearchTime = Tvalues[0];
                SupportTime = B_SearchTime - T_SearchTime;
                return SupportTime;
        }
        else
        {
                return error;
        }
}
```

6.4. Summary

This chapter provided a first step towards validating the EAGQ method. Firstly, the method was applied on the ArchiSurance case study which proved that applying the method is feasible and that the combination of different measurement and analysis techniques in the EA level and the goal level is possible. More specifically, the incorporation of cost analysis and performance analysis in the same case was demonstrated. Secondly, an indication of the fact that the EAGQ method can be implemented in a tool was also provided through a small set of algorithms. The algorithms support the propagation and calculation of the indicator values across the PI structure as part of Step I. In order to provide further validation on the method, a set of interviews with experts was conducted for evaluating the method. This approach for validating the EAGQ method is discussed in the next chapter.

CHAPTER 7

7. Validation of the EAGQ Method

This chapter presents the evaluation of the EAGQ method. After applying the method on the ArchiSurance case and demonstrating that the method is able to provide valuable analysis results for this case, a series of interviews were held for validating the method by experienced practitioners. The setting of the evaluation process, the criteria examined through the interviews and the results of the validation are discussed in the following sections. The last section provides the personal reflection of the author on the EAGQ method.

7.1. Evaluation method & Assessment Criteria

The third phase of the Design Cycle described in Section 1.5 comprises the 'Design Validation'. In this phase, the expected value of the designed artifact and its contribution to the initial goals of the research are examined. The focus of the validation conducted for this research focuses on the two first types of knowledge questions introduced by Wieringa [14] [15], which cover the expected effects and the expected value of the method. The 'expected effect' related questions aim to address whether the method solves the problems that drove the research and the expected value questions aim to assess the method against a set of criteria. The criteria which were selected for assessing the quality and value of the method are:

- Complexity or ease of use
- Ease of understanding
- Usefulness
- Practical use or applicability

The assessment of these criteria was opinion-based. The method that was used for evaluating the EAGQ method consists of a set of semi-structured interviews with five practitioners in the field of EA and Strategy. All participants came from BiZZdesign, the company in which this research was carried out. The participants have different roles. More specifically, their roles are: Senior Research Consultant, EA Consultant (two participants with this role), Product Manager and Junior Research Consultant (who is also a Strategy PhD Researcher). All the participants have experience with the ArchiMate® modeling language, while four out of five professionals have performed, have been involved or are familiar with customer projects which included quantitative analysis of the EA models.

The semi-structured interview setting was selected because the interviewer had only one chance of performing the interview with the professionals and because of the ability of this type of interview to provide reliable and comparable results.

Separate interview sessions were held with each one of the participants. Each interview session lasted approximately an hour and it included two parts; a presentation and a discussion based on a set of questions. The presentation lasted 25-30 minutes and in the remaining 30 minutes the participant had the chance to reflect on his/her perception of the method and provide feedback with respect to the presented information. The presentation covered the following topics:

- Research goals and objectives
- Motivation of the research and goal of the method
- Description and Explanation of the EAGQ method
- Application of the method through the ArchiSurance case study

The questions which were asked in order to evaluate each of the four selected criteria are presented in Table 73. When all three sets of questions were answered, the participants were requested to engage in an open discussion and provide an overall comment or a general feedback on the method.

Criterion	Questions
Complexity or Ease of use	1. Do you think you could apply the EAGQ method based on the presentation? 2. If you would use it, do you think it would be easy to use?
Ease of understanding	3. Do you consider the following steps clear and understandable? • Step C: Identify indicators and measures • Step D: Build the PI Structure • Step E: Perform the measurements in the EA model • Step I: Perform indicator analysis for every target EA alternative
Usefulness	4. Do you consider the EAGQ method useful? • If yes, in which areas do you find it more useful? ○ E.g. Decision-making, EA project investment, design decisions, goal quantification 5. Do you think that the EAGQ method can provide more insight to boardroom stakeholders with respect to goal evaluation?
Practical use or applicability	6. Do you think the method's *outcome* has value to customers? If yes, how? If not, why? 7. What do you think of the method with respect to effort to apply it? • Think of: ○ Initial effort vs. reusability of the PI structure ○ Effort vs. benefits
Open discussion	1. Do you have any further comments regarding the method? • Recommendations for improvement? • Feedback? 2. Do you think that all necessary activities are included in the method? If not, which should be added or changed?

Table 73 - Evaluation criteria and Questions

7.2. Validation results

In this section, the results of the validation sessions are discussed for each one of the four criteria. The general comments that were given by the participants in the validation are also provided in the end of the section.

Complexity or Ease of use and Ease of understanding

Most of the practitioners (four out of five) considered the method as feasible and doable and considered themselves as capable to apply it when the presentation was finished. All of them, though, stated that there were a lot of information in the presentation, which necessitates a second look on the presentation and the details of the more technical steps regarding the measurements in order to be able to apply it themselves. The examples provided throughout the presentation, which were derived from Chapter 6, were considered helpful for guiding the application of the method by the Junior Research Consultant, while one of the EA Consultants based his answer regarding his ability to apply the method (first question) on his past experience of applying analysis techniques in the EA-level.

From the perspective of understanding the EAGQ method, the answers covered two aspects. Firstly, the practitioners stated that, in a high-level, the method was understandable and that they were able to follow the presentation. Additionally, they mentioned that the steps made sense and it was a common comment from the Senior and Junior Research Consultant and one of the EA Consultants that the meaning of the steps and the line of reasoning behind them were clear. Furthermore, the fact that activities, such as Activity D1 for eliminating duplicate measures, which may seem intuitive or redundant for some users but not for others due to not being used in this kind of logical thinking or due to lack of experience, was also viewed as a positive aspect of the EAGQ method by the Junior Research Consultant. Secondly, the participants expressed their confusion on a few activities, so on a more detailed level of analyzing the method. The activities that puzzled most practitioners and the comments on these activities are the following:

- **Activity B2 – Relate the Baseline EA concepts with the Requirements.** Three out of five practitioners (Product Manager, Senior Research Consultant and EA Consultant) were puzzled with the use of the 'association' relation between EA concepts and requirements. Their main point was that these relations can be derived as indirect relations by following the traceability links and by using the relations that already exist in the EA model. Thus, while they found the purpose of introducing the difference between the relationships justified, i.e. expressing a potential gap between goals and the architecture in Activity B3, they were skeptical about the complexity and effort of introducing double relations between requirements and EA concepts. This observation is correct and, as suggested by the Product Manager, the solution could be the assignment of only association relations in a requirement and not both relation types in order to simplify the models.
  A similar discussion was made for the **Activity C4** and the design of the **goals' focus areas**. The use of the traceability links can enhance the automation of this activity as well.
- **Activity C1 – Ask questions for the lowest-level goals to reveal the measurement needs.** How the questions are derived was the main question regarding this activity. While the practitioners agreed that this is an easy task for more concrete hard goals, the user of the EAGQ method could encounter problems with softer goals. The explanation of the GQM and GQ(I)M approach as well as the addition of preparatory surveys or driver assessments clarified the process for deriving the questions. One of the EA consultants considered a pre-determined large questionnaire from which the user can select questions that best apply to a particular case as an alternative for identifying the right set of questions.
- **Activity D3 – Build the PI Structure.** The main clarification that most of the practitioners asked regarding this activity was whether the goal model and the PI structure have always the same structure. In the case of 1-to-1 correspondence between goals and indicators, the Senior Research Consultant and the Product Manager proposed a different way of representing the indicators in order to reduce the complexity by drawing a separate model, such as modeling the *indicators* as *attributes of goals* and *measures* as *attributes of requirements*. All practitioners, though, agreed that designing the PI Structure is not a difficult task and that it

improves understanding or analyzing of the EAGQ method, especially in the context of the thesis.

- **Activity E2 – Compute the measures.** The decision on the most appropriate measurement method was also discussed by one of the EA Consultants and the Product Manager. While for cost analysis, cost allocation may seem an easy propagation of values, in case the allocation takes into account other factors, such as utilization of resources, may become more complex. Additionally, other types of financial analysis may increase the complexity and the difficulty of the measurements; while measuring qualitative factors, on the other hand, may require guidance for deciding on the measurement approach. In general, though, the cost and performance analysis is perceived as a good way to start due to the quantitative and solid nature of the related techniques.
- **Activity I2 – Calculate indicator performance levels.** The clarifications that the two EA Consultants asked in this activity were regarding the use of the 'time-difference' multiplier. One's intuition was that the time-difference shows how the performance levels change or "move" in time, while the other thought of it in terms of uncertainty due to the time in the future that the target EA is going to be realized. So, an EA alternative realized further in the future than another one would mean more uncertainty and consequently lower performance levels. None of the practitioners had a problem or disagreement with the actual use of the multiplier in the EAGQ method.

As far as the ease of applying the EAGQ method is concerned, all of the practitioners had a difficulty in assessing it. While they consider the method logical as well as the activities included in the method meaningful, they characterized the method as complex because of its size. Ten steps and an average of 2.6 activities per step cause an increased perceived difficulty, because "it looks a lot", as the Senior and Junior Research Consultants said, even if the steps are not difficult themselves. In order to assess the ease of applying the method, the practitioners raised a number of practical issues which influence the usability of the method. These issues are discussed below, where the criterion of 'Practical use or applicability' is analyzed.

Usefulness

The answers regarding the usefulness of the EAGQ method were aligned. All practitioners considered the method as useful and valuable especially in the areas of decision-making and, consequently, EA project investment. The Senior Research Consultant viewed the goal quantification more as a means to realizing the decision-making process.

The practitioners believe that the outcomes of the method are useful for comparing and deciding between alternatives and for evaluating which projects should be continued, changed or canceled. The analysis of which EA elements contribute to certain goals can influence deciding about projects which aim to change these elements, for instance. Moreover, one of the EA Consultants stated that there is a gap in quantifying the value of EA and taking decisions based on such quantitative analysis in practice. This gap could be filled in with a method like the EAGQ method. The Senior Research Consultant also proposed that the EAGQ method can be used in all areas where EA is used as a management instrument and it can actually promote EA as an instrument for supporting decision-making. Furthermore, the

Junior Research Consultant pointed out that the method could also help in making business models, since it can support the identification of the cost structure of the architecture. This observation was related to the paper by Iacob et al. [116], where EA models are used for driving changes in business models.

The Product Manager and one of the EA Consultants also expressed their concerns regarding characteristics of the method which might limit its usefulness. The Product Manager was not sure whether the method is simple and fast enough to enable its incorporation in the decision-making cycle and the project-investment cycle. The user has to make sure that the goals can be translated to indicators which should be derived through measurements in the EA models which, in turn, should include all the data, before the stakeholders move to the next cycle. The EA Consultant, on the other hand, compared the method to business case analysis and stated that the absolute values calculated through the method should not be perceived as exact numbers but more as an indication of the expected numbers and of course as a means for comparing the alternatives. The uncertainty that these absolute values carry due to the time in the future aspect can give misleading expectations if this factor is not taken into account. This was also the opinion of the Senior Research Consultant. As an overall comment, though, he believes that a rigorous and thorough method to evaluate alternatives is needed.

When the practitioners had to evaluate the method with respect to boardroom stakeholders, all had the same initial reaction: *The outcomes of the method are useful, but perhaps not the method itself*. The reason behind this statement is that the boardroom stakeholders are interested in the results of an analysis that can help them make decisions, but they do not want to know how these results were retrieved. What they would like to know, though, is whether they can trust the results. So, for these stakeholders it is sufficient to know that the applied method is a valid and solid approach which can give them some kind of evidence and a feeling that the results make sense and are trustworthy. Thus, the selection of such a method should be verified by specialists in the organization before the analysis starts, as it was stated by one of the EA Consultants. A very different approach came from the Product Manager, though, who wondered whether the same analysis outcomes could be achieved without using the EA models. To conclude, while the method is mostly addressed to architects, portfolio managers or other specialists (stated by an EA Consultant), the outcome extracts from this complexity and is simple and in terms that management can understand (stated by the Junior Research Consultant).

Practical use or applicability

The answers to the first question regarding practical use of the EAGQ method were quite contradicting. On the one hand, there were practitioners who were very positive about the application of the method from the customers' perspective, while, on the other hand, there were also practitioners who were more skeptical or even discouraging about the method's success in practice.

The first group, which is the majority (four out of five), considers the method as valuable to customers. Three out of four think that the method can guide the organizations in decision-making and it can promote the value of EA modeling, especially in profit-oriented organizations, since the method can demonstrate how such models relate to revenues or sales, for example. One of practitioners also pointed out which would be the most suitable organizations to use such a method. Companies with a mature level of EA could be the early adopters of the method and the ones which could benefit more from it, since the quality of their EA models is better which is a requirement for applying the EAGQ method (stated by an EA Consultant). The forth practitioner of this group was also positive but he pointed out that the effort of applying the method is an important factor.

The last practitioner (Product Manager), who was more discouraging, agreed that there is an interest in the research community regarding analysis in the EA-level, but he does not think that the same interest exists in practice. Of course, there are organizations which have developed their own methods, but not many other organizations try to copy this effort. His overall comment, though, was that the method's outcomes can be valuable for driving business value or showing business value of EA, but under certain circumstances.

Regarding the effort to apply the method, all practitioners share the same opinion. The effort, and especially the initial effort, is high. The Product Manager and the Senior Research Consultant questioned the reusability of the PI structure or part of the structure. Their uncertainty resides on organizations with flexible goals which change very often. In such cases, as it is pointed out by the Product Manager, the rework is high for redefining and adjusting the indicators and the measurements. For these organizations, the benefits may also be limited because of the need of up-to-date, real-time data for decision-making which may make the PI structure difficult to maintain. In case the EA of the organization does not change drastically, as discussed by the Junior Research Consultant, the reusability is higher and, hence, it could justify the initial effort. Other factors that can affect the decision of an organization to invest in this method are how well the company's data is organized, how mature its EA is, how big is the amount of money involved in the organizational change (the higher the amount, the higher the willingness to make the effort) and how much the organization relies on an organized and structured approach for decision making comparing to simple insight or other political reasons, for instance. Thus, while there are benefits from applying the EAGQ method, the decision to apply it depends on practical issues and characteristics of the organization.

The practical issues that influence the adoption of the EAGQ method, as described by the practitioners throughout the interview sessions, are the following:

- Regarding Activity C2 (Assign indicators to each goal), the Product Manager doubted the ease of *assigning values to the threshold, worst and tolerance attributes of indicators*. He identified as the source of the problem the fact that the people who are the owners of such information are different from the people that work on the EA models. Hence, the *involvement of many people in a variety of roles* may hinder the application of the method. This observation aligns with the uncertainty of the data correctness pointed out by most of the practitioners.
- *Quality of data* and *data correctness* are factors that have a strong influence on the decisions made in real life. Many organizations have a lot of data in different parts of the organization with different levels of quality. Including the uncertainty of the quality of input data in the method could make the stakeholders aware of the uncertainty that the outcomes of applying the method bare and, hence, avoid making decisions based on the perception that the outputs comprise strong and certain facts. The management of data with different levels of quality is also a practical issue related to master data management.

- Most organizations do *not have the appropriate information in their models* in order to derive measurements from the EA models. Thus, someone needs to populate the model with the necessary data. If this is established, then the models could become very rich and the analysis of them could become easy. The road to this situation is not easy, though, because of the people involved as well.
- The *EA models* in practice are *not completely correct*, mostly in terms of relationships which are either wrong or missing. The correctness of the baseline EA model might influence the measurements derived from it.
- The existence of good *tool support* can also affect the adoption of the method. The tool support can help in automating the design of the PI Structure due to the 1-to-1 correspondence, the calculation of measures and indicators and of course the derivation of the models regarding the requirements' realization degree (categorization) and the goals' focus area. The automation, of course, includes a lot of initial effort in order to develop, for example, the scripts for the computation of the indicators.
- The baseline architecture is *not static* and the target architecture comprises a *moving target* as well. Thus, there is an uncertainty imposed by changes in the environment which is difficult to incorporate in the method or the analysis. To improve the expected benefits from applying the method, the organizations could use the outcome as an indication towards a direction to the future. Assessing the transitions from the baseline to the target model, apart from assessing the target architecture as a predetermined goal, can increase the applicability of the method and the certainty of the outcomes.

Overall comments and Suggestions for improvements or changes

The EAGQ method is considered as a sound and thorough method which has been worked out in depth and in detail through the examples as well. This opinion was the most prevalent one among the participants during the validation of the method. One of the EA Consultants also characterized the method as very impressive, while the Junior Research Consultant stated that the method is definitely useful because it links two different domains and it helps in making educated decisions. There were of course suggestions for improving the method, which are the following:

- **To make the method smaller and simpler.** While the Senior and Junior Research Consultants believe that all necessary activities are included and they could not see something that should or could be omitted, they suggested that making the method smaller would be an improvement. An idea was to consolidate, if possible, some of the activities. This viewpoint could be combined with the opinion of the Product Manager who proposed to skip the design of duplicate models. To make the method simpler, the issue of the association relations in Activity B2 should be solved and the use of the traceability links in a larger extent should be promoted.
- **To determine, in advance, the stakeholders for applying the method and performing the analysis.** One of the EA Consultants proposed that the involvement of various people in different roles throughout the method should be established before the application of the method. This could be achieved by introducing an initial step for

defining the focus of the project and the related stakeholders from the organizational perspective.

- **To incorporate manuals and standard sets of indicators and/or measures.** Both EA Consultants consider important to support the method with manuals, which will provide guidance for applying the method in practice, as well as with a pre-defined set of indicators and/or measures which the user can use as a starting point and can then expand by adjusting it to specific cases.
- **To add a step for performing validity checks on the EA models.** One of the EA Consultants suggested that performing a sanity check on the correctness or validity of the EA models and evaluating which elements and relations in the EA models are necessary, would help in getting more correct and trustworthy outcomes. Additionally, it will force the architects to improve their EA models as well.
- **To invest time in Step J for identifying the most meaningful and suitable ways for presenting the outcomes of the method.** Both EA Consultants value the right presentation of the outputs of the method. Since the decision-makers are going to view only the results, the way the results are presented makes a difference. Different people with various backgrounds participate in the decision-making process and these people may have different preferences or understanding on presentation methods, e.g. graphs, charts, tables, documentation. Their opinion and preferences should be taken into account before presenting the results.
- **To add a step for re-evaluating and adjusting the decisions and the models.** In practice, as one of the EA Consultants stated, there is a trade-off between the quality of a decision and the timeliness of the decision. To be able to take decisions faster, but also keep the quality of the decisions high, an update cycle is needed. The update cycle comprises updating data and the corresponding measurements as well as assessing the value of the project, and based on these assessments, the goals, EA models and the projects should be re-evaluated and changed or canceled. In that way, more room would be made for other (new) projects that can provide more value. The EAGQ method could help in realizing this update cycle and re-evaluating by putting in the method the data again and again and, if necessary, re-adjusting the goals and the EA models. Therefore, the suggested improvement would be the addition of an iterative feedback loop.

Table 74 summarizes the opinions provided by the practitioners throughout the validation sessions, which were discussed above.

7.3. Summary and Personal reflection on the EAGQ Method

The points made by the participants in the validation process of the EAGQ method are justified in general. The EAGQ method is a resource-intensive method to be applied in a company which does not have a well-organized structure of its data. For an organization to be able to use the method and expect benefits in return in a short time, the maturity level of its EA should be also high. These two factors along with the prerequisite of the data to be accurate and the people involved ready to trust a quantitative analysis through EA models are also crucial.

As a personal opinion, the author believes that the EAGQ method can provide more secure decision-making and can assist in the selection of the most beneficial EA model. The outcomes of the method, i.e. the indicator structure and the values of indicators and goals, can be of value for the organizations, since they provide a more solid insight on the assessment of goals and evidence for supporting the qualitative values assigned to goals in the end of the process.

The large number of steps and activities indicates that performing an analysis that spans the EA and the goal models involves a great deal of effort and a variety of people to participate. The fact, though, that the high-level business goals of an organization are less volatile and that the EA of an organization does not change drastically can facilitate the reusability of the indicator structure and can assist in the reduction of effort of using the method across time. To improve even more the experience of using the method, the implementation of the method in a tool would be crucial.

Criteria	Summary of answers
Complexity or Ease of use	The method is not difficult per se, but the number of steps and activities increase its complexity and perceived difficulty. Practical issues can also influence the complexity of applying the EAGQ method, which was the reason that the practitioners had a difficulty in assessing the method's actual 'ease of use' at the moment.
Ease of understanding	The method is not difficult to understand and to follow and each of the activities is meaningful. Some of the activities were further discussed because of: • the involvement of judgment of the user for selecting or deciding about e.g. questions, measures, measurement methods, etc. Guidance should be provided while performing this decision-making process. • the current way of using traceability links and deriving indirect relations in EA models in practice • intuitive understanding of a concept (term)
Usefulness	The method in general is useful and it can contribute in decision-making and comparison of alternatives. Concerns exist regarding its alignment with decision-making cycles and regarding the uncertainty that the absolute outcome values carry. From the perspective of boardroom stakeholders, the outcome of the method is useful for them, but the method itself might not be. What is important is to ensure the stakeholders that the method used for the analysis is valid and trustworthy.
Practical use or applicability	The outcomes of the method can be valuable for organizations. Some practical issues, though, might hinder the success of the method's adoption in real life. The initial effort for applying the method is also quite high, which enhances the chances of the method being adopted by organizations with mature EA models, large investments and a more structured mindset for deciding upon organizational change.

Table 74 - Validation outcome summary per assessment criterion

CHAPTER 8

8. Contribution, Recommendations, Limitations & Future Work

This chapter provides an overview of the research conducted for this thesis project and recommendations for practitioners as guidelines for benefiting from the method. Additionally, it discusses the limitations of the method and the research, and proposes how the research on the topic can be continued.

8.1. Answers to Research Questions

RQ1. Why are quantifiable assessments of EA and Goal models needed?

The main identified problems that guided and initiated this research were:

- Lack of relationship between existing analysis techniques at the EA and goal level
- Difficulties in building EA with respect to assessment of organizational change. While EA is used in order to address organizational change, there is a difficulty or inability to assess organizational change by assessing the EA.
- Difficulties in measuring the impact of EA, which relate to the following areas:
    - Decision-making for EA projects
    - Assessment of expected benefits
    - Assessment of goal values
    - Assessment of EA project resources
- Difficulties in demonstrating the value of EA to high-level management who is mainly involved with the goal level.

RQ2. What are the existing quantification techniques/models/mechanisms for expected cost and benefit for

a) **Goal models?**

b) **Architecture models?**

A broad literature review has been conducted and a large number of techniques has been identified which aim to quantify or assess goals as well as their interrelations. Some of these techniques also provide ways to transform high-level business goals into measurable goals in order to facilitate their quantification or assessment.

The literature review with respect to the EA models is also vast. It covers financial analysis techniques used for evaluating IT-related projects, methods for cost allocation, approaches for identifying and specifying costs and benefits as well as performance-related and availability-related measures. Some of these techniques were specific for EA models, others have been used in analyzing EA models, but the majority of them has not been applied in the field of EA.

RQ3. Extend ARMOR with quantification mechanism(s).

To address this research question, the EAGQ method was designed. This method aims to quantify the contribution of EA with respect to business goal changes. It is a method of ten steps, which range from building the goal and EA models and measuring various EA models against the goals, to analyzing these measurements and evaluating the value provided by changes in the EA level to the target goals. To enable the analysis, a new model is introduced, the Performance Indicator (PI) Structure, which contains indicators and measures. The indicators correspond to goals, while the measures to requirements. A more volatile set of measures, i.e. the component measures of the measures corresponding to the requirement-level, comprise the bottom-level of the PI Structure. The component measures are derived from the baseline and target EA models. The information flow in the method originates from EA concepts, from which information is derived, to the component measures and then across the PI Structure in a bottom-up propagation of values based on pre-defined calculation formulas. The overall performance indicator values are used as goal satisfaction values for comparing different EA alternatives.

ARMOR is the language based on which the ArchiMate® Motivation Extension was developed. The introduction of the EAGQ method extends the language in terms of goal attributes and meaning of relations between goals; enriches the traceability links between motivation extension concepts and EA concepts; links the analysis in the two domains; and provides a separate model via which the goal model can be analyzed and evaluated.

RQ4. Design the implementation of the quantification mechanism(s) in the Architect tool.

The implementation of the EAGQ method has not been carried out for the whole method, but an algorithm is provided in order to demonstrate that it is feasible to automate the method with the use of tool support. The focus of the operationalization is upon the propagation of values in the PI Structure in order to enable automated analysis of indicators and measures. The attributes specified for the indicators and measures while applying the EAGQ method can be modeled as properties of these concepts in a tool. Additionally, the calculation formulas can be modeled as separate functions or methods.

RQ5. What is the *contribution* of the quantification mechanism(s)?

Based on the validation of the method through interview sessions with practitioners and as a personal evaluation by the author through applying the method on the ArchiSurance case study, the EAGQ method is believed:

- to support a more educated and more structured decision-making process
- to provide evidence for the qualitative analysis and evaluation of goals through quantifications in the EA level
- to provide a link between the goal and EA analysis domains
- to promote EA as a tool for analyzing goals and decision-making
- to integrate separate analysis methods and outcomes that, alternatively, would have been performed for each goal independently. The integration of the analysis provides a more complete and holistic picture where the influence of the design choices can be evaluated as a whole. The degree of contribution of these decisions/choices to individual goals and focus areas can also be assessed.

- to encourage more careful design and maintenance of EA models as well as the performance of validity checks to improve the quality of EA and to support the EA-level analysis and measurements.

As for the companies that the EAGQ method addresses best are the ones with mature EA models, high availability of data and engagement with investment decisions which involve large amounts of expenses. The applicability of the method in this type of companies is a result of the practitioners' judgment during the validation phase of this research.

8.2. Recommendations

The interviews, which were conducted with five practitioners in order to validate the EAGQ method, provided a lot of feedback and resulted in a set of suggestions for supporting the method's adoption by organizations. The practical recommendations are:

- **Use of proof of concept.** When introducing the method, use an example with a small scope in an area where data is available, in order to demonstrate the value of the method and what can be achieved.
- **Present the essence of the method to companies and adjust the method based on the company.** Focusing on the benefits offered by the EAGQ method and what the user can achieve by applying the method should be emphasized before showing all steps in detail to engage the user in valuing the method. Additionally, steps that are optional could be omitted depending on the case that the method is going to be applied on.
- **Determine, in advance, the stakeholders for applying the method and performing the analysis.** See Section 7.2.
- **Scope the analysis and design the models that are essential for the measurements in the EA level.** In that way, the effort can be reduced and the quality of the EA models used for deriving the measures can be improved.
- **Invest some time for assessing the needs of the organization before selecting the measures and indicators, in order to identify the core problem that each goal tries to address.** High-level business goals should be refined appropriately and with special care so that their dependencies can be expressed in a computational manner. For qualitative or soft goals, a further investigation, survey or more thorough analysis can assist in determining the measurement approach at the EA level and of course whether such a measurement would be feasible to be performed through the EA model.

8.3. Limitations

Research Limitations

The limitations regarding the research conducted in this thesis project are the following:

1. The available literature in the field of goal analysis is quite extensive and a number of techniques exist for relating goals with software architectures, but research on goals with respect to Enterprise Architecture was not found apart from two methods regarding strategic alignment.
2. The evaluation of goals and benefits in the goal level as well as the evaluation of influences among goals are mostly qualitative. This observation means that either

goals are not expected to be evaluated quantitatively in organizations or that quantitative analysis is done in organizations but not shared among the research community.

3. The literature regarding EA-specific analysis techniques is also limited. The change analyzed in EA models is related to IT-driven and technology-driven changes. While organization culture or structure changes can be expressed through the business layer of the ArchiMate® modeling language, no analysis methods have been found in this area. Thus, goals related to such changes might not be measurable through EA models.
4. The interviews conducted for validating the EAGQ method provide insight on *perceived* ease of use, understanding, usefulness and applicability. Time constraints and not implementing the method in a tool limited the testing of the method.

Practical Limitations

In practice, a number of factors can influence the adoption and successful use of the EAGQ method. These factors have been revealed through the interview sessions and discussed in Section 7.2. The most important, in short, are:

- Uncertain quality and correctness of data
- Lack of integration of data in EA models
- Uncertainty due to time distance between the baseline and target EA and the fast changing environment
- Uncertainty about the correctness of EA models and the validity of relationships in the models
- Need for judgment calls, which reduces the degree of operationalization of the method
- Involvement of various people with various roles during the application of the method, which raises an effort for organizing the collaboration of these people

Additionally, since the analysis could be more effective and easier when cost and performance techniques are used and cost-related and performance-related measures are identified, the applicability of the EAGQ method on non-profit organizations may be limited. This observation resides on the fact that non-profit organizations focus on softer goals which are not that easy to quantify through information from the EA models.

8.4. Future Work

The focus of future work should be to address the limitations and to assist the realization of the recommendations in order to improve the method and enhance its success in real-life. The first step towards this direction will be to empirically test and evaluate the criteria of 'ease of use' and 'ease of understanding' by asking people to apply the method themselves and check the accuracy and quality of the output. Another type of validity testing could focus on verifying the notion of re-usability of the method, i.e. the fact that by repeating the method, the application effort is reduced and the steps become smaller and less complex.

Moreover, an evaluation of the steps of the method should take place, in order to assess which steps can be integrated or omitted, if any. This activity could result in reducing the number of steps, which were identified as many by the practitioners. Additionally, how the method could be embedded in the decision-making and investment processes or cycles should be investigated, as well as the integration of the method with existing tools the companies use for modeling.

Finally, a set of more practical suggestions and additions for improving the method includes:

- **Incorporating manuals and standard sets of indicators and/or measures.** Manuals of applying the method, standard sets of indicators and measures, and a general questionnaire with questions for identifying the measurement needs of goals can contribute in the applicability of the method.
- **Adding a step for performing validity checks on the EA models.** See Section 7.2.
- **Providing tool support for automating the design of the PI structure and the propagation of indicator values.**
- **Providing alternatives to organizations with respect to the PI Structure.** When the method will be supported by a tool, in case there is 1-to-1 correspondence between goals and indicators, the option of not drawing the indicator structure as a separate model can be provided. Such an option would include incorporating the indicators and measures in the goal model. Indicators can be modeled as 'attributes' of goals and measures as 'attributes' of requirements.

References

[1] M.-E. Iacob, H. Jonkers, D. Quartel, H. Franken, and H. van den Berg, *Delivering Enterprise Architecture with TOGAF and ArchiMate*. BiZZdesign Academy, 2012, pp. 1–300.

[2] M.-E. Iacob and H. Jonkers, “Quantitative analysis of enterprise architectures,” *Interoperability of Enterprise Software and Applications*, pp. 239–252, 2006.

[3] T. Tamm and P. Seddon, “How does enterprise architecture add value to organisations?,” *Communications of the Association for Information Systems*, vol. 28, no. 1, pp. 141–168, 2011.

[4] W. Engelsman and R. Wieringa, “Goal-Oriented Requirements Engineering and Enterprise Architecture : Two Case Studies and Some Lessons Learned,” in *18th International Working Conference, REFSQ 2012 in Requirements Engineering: Foundation for Software Quality*, no. iv, Springer Berlin Heidelberg, 2012, pp. 306–320.

[5] L. S. Rodrigues and L. Amaral, “Issues in Enterprise Architecture Value,” *Journal of Enterprise Architecture*, vol. 6, no. 4, pp. 27–32, 2010.

[6] B. van der Raadt, S. Schouten, and H. van Vliet, “Stakeholder Perception of Enterprise Architecture,” in *Software Architecture*, Springer Berlin Heidelberg, 2008, pp. 19–34.

[7] D. Quartel, W. Engelsman, H. Jonkers, and M. Van Sinderen, “A Goal-Oriented Requirements Modelling Language for Enterprise Architecture,” *2009 IEEE International Enterprise Distributed Object Computing Conference*, pp. 3–13, Sep. 2009.

[8] W. Engelsman, D. Quartel, H. Jonkers, and H. Franken, *Business Requirements Management*. BiZZdesign Academy, 2010, pp. 9–122.

[9] M. L. Markus, “Technochange management: using IT to drive organizational change,” *Journal of Information Technology*, vol. 19, no. 1, pp. 4–20, Mar. 2004.

[10] J. Karlsson and K. Ryan, “A cost-value approach for prioritizing requirements,” *IEEE Software*, vol. 14, no. 5, pp. 67–74, 1997.

[11] A. van Lamsweerde, “Goal-oriented requirements engineering: A guided tour,” *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium*, vol. IEEE, pp. 249–262, 2001.

[12] F. Crundwell, *Finance for Engineers: Evaluation and Funding of Capital Projects*. Springer-Verlag London Limited, 2008.

[13] J. Peppard, J. Ward, and E. Daniel, “Managing the Realization of Business Benefits from IT Investments,” *MIS Quarterly Executive*, vol. 6, no. 1, 2007.

[14] R. Wieringa, “Design science as nested problem solving,” in *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology - DESRIST ’09*, 2009, pp. 1–12.

[15] R. Wieringa and A. Morali, "Technical action research as a validation method in information systems design science," *Design Science Research in Information Systems. Advances in Theory and Practice*, pp. 220–238, 2012.

[16] V. Kavakli and P. Loucopoulos, "Goal-driven business process analysis application in electricity deregulation," *Information Systems*, vol. 24, no. 3, pp. 187–207, May 1999.

[17] The Open Group, *ArchiMate 2.1 Specification*. Van Haren Publishing, 2013.

[18] W. Engelsman and R. J. Wieringa, "Understandability of Goal-Oriented Requirements Engineering Concepts for Enterprise Architects," in *Advanced Information Systems Engineering (CAiSE), 26th international conference*, 2014.

[19] R. C. de Boer and H. van Vliet, "On the similarity between requirements and architecture," *Journal of Systems and Software*, vol. 82, no. 3, pp. 544–550, Mar. 2009.

[20] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," *Science of computer programming*, vol. 20, no. 1, pp. 3–50, 1993.

[21] R. Ali, F. Dalpiaz, and P. Giorgini, "A goal-based framework for contextual requirements modeling and analysis," *Requirements Engineering*, vol. 15, no. 4, pp. 439–458, Jul. 2010.

[22] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An Agent-Oriented Software Development Methodology," *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, May 2004.

[23] V. M. B. Werneck, A. de P. A. Oliveira, and J. C. S. do Prado Leite, "Comparing GORE Frameworks: i-star and KAOS.," *WER*, 2009.

[24] E. Kavakli, "Goal-Oriented Requirements Engineering: A Unifying Framework," *Requirements Engineering*, vol. 6, no. 4, pp. 237–251, Jan. 2002.

[25] R. Darimont and M. Lemoine, "Goal-oriented Analysis of Regulations.," *ReMo2V*, pp. 838–844, 2006.

[26] J. P. A. Almeida, M.-E. Iacob, and P. Eck, "Requirements traceability in model-driven development: Applying model and transformation conformance," *Information Systems Frontiers*, vol. 9, no. 4, pp. 327–342, Aug. 2007.

[27] A. Tang, J. Han, and P. Chen, "A Comparative Analysis of Architecture Frameworks," in *Proceedings of the 11th Asia-Pacific Software Engineering Conference (ASPEC'04)*, 2004, pp. 640–647.

[28] M. Iacob and H. Jonkers, "Quantitative Analysis of Enterprise Architectures," *Interoperability of Enterprise Software and Applications*, pp. 239–252, 2006.

[29] D. Chen, G. Doumeingts, and F. Vernadat, "Architectures for enterprise integration and interoperability: Past, present and future," *Computers in Industry*, vol. 59, no. 7, pp. 647–659, Sep. 2008.

[30] H. Jonkers, M. Lankhorst, R. van Buuren, S. Hoppenbrouwers, M. Bonsangue, and L. van der Torre, "Concepts for modeling enterprise architectures," *International Journal of Cooperative Information Systems*, vol. 13, no. 3, pp. 257–287, 2004.

[31] M. Lankhorst et al., *Enterprise Architecture at Work: Modeling, Communication and Analysis*, Third Edit. Springer-Verlag Berlin Heidelberg, 2013.

[32] M. M. Lankhorst, "Enterprise architecture modelling—the issue of integration," *Advanced Engineering Informatics*, vol. 18, no. 4, pp. 205–216, Oct. 2004.

[33] R. van Buuren, S. Hoppenbrouwers, H. Jonkers, M. Lankhorst, and G. V. van Zanten, "Architecture Language Reference Manual," 2006.

[34] BiZZdesign, "BiZZdesign Architect: Leading Software for Enterprise Architecture." [Online]. Available: http://www.bizzdesign.com/tools/bizzdesign-architect/. [Accessed: 06-Apr-2014].

[35] IFEAD, "Enterprise Architecture Tools." [Online]. Available: http://www.enterprise-architecture.info/EA_Tools.htm. [Accessed: 06-Apr-2014].

[36] A. Jansen, T. de Vries, P. Avgeriou, and M. van Veelen, "Sharing the architectural knowledge of quantitative analysis," *Quality of Software Architectures. Models and Architectures*, vol. Springer B, pp. 220–234, 2008.

[37] A. Abran, J. Desharnais, and J. J. Cuadrado-gallego, "Measurement and quantification are not the same : ISO 15939 and ISO 9126," no. August 2010, pp. 585–601, 2012.

[38] H. Jonkers and M.-E. Iacob, "Performance and Cost Analysis of Service-Oriented Enterprise Architectures," in *Global Implications of Modern Enterprise Information Systems: Technologies and Applications*, A. Gunasekaran, Ed. IGI Global, 2008, pp. 49–73.

[39] F. Matthes, S. Buckl, J. Leitel, and C. M. Schweda, *Enterprise Architecture Management Tool Survey 2008 About sebis*. Technische Universität München, 2008.

[40] S. Buckl, F. Matthes, and C. M. Schweda, "Conceptual Models for Cross-Cutting Aspects in Enterprise Architecture Modeling," *2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops*, pp. 245–252, Oct. 2010.

[41] G. Schwetje and S. Vaseghi, "12 Finance," in *The Business Plan: How to Win Your Investors' Confidence*, Springer-Verlag Berlin Heidelberg, 2007, pp. 107–145.

[42] B. Mutschler, *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*. Enschede, The Netherlands, 2008.

[43] Jaap Schekkerman, *Enterprise Architecture Good Practices Guide*. Victoria, BC, Canada: Trafford Publisher, 2008, pp. 1–212.

[44] Wikipedia, "Expense." [Online]. Available: http://en.wikipedia.org/wiki/Expense. [Accessed: 09-Apr-2014].

[45] D. G. Woodward, “Life cycle costing—Theory, information acquisition and application,” *International Journal of Project Management*, vol. 15, no. 6, pp. 335–344, Dec. 1997.

[46] R. G. Slot, *A method for valuing architecture-based business transformation and measuring the value of solutions architecture*. 2010, p. 184.

[47] The Open Group, *TOGAF Version 9.1*, First Edit. Van Haren Publishing, 2011, pp. 1–615.

[48] R. L. Vienneau and D. Nicholls, “Develop and Maintain a Life-cycle Business Case,” 2008.

[49] J. Ward, E. Daniel, and J. Peppard, “Building better business cases for IT investments,” *MIS Quarterly Executive*, vol. 7, no. 1, pp. 67–78, 2008.

[50] S. M. Eckartz, “Managing the Business Case Development in Inter-organizational IT Projects: A Methodology and its Application.” Centre for Telematics and Information Technology, University of Twente, 31-Aug-2012.

[51] W. Harrison, D. Raffo, J. Settle, and N. Eickelmann, “Technology Review : Adapting Financial Measures : Making a Business Case for Software Process Improvement,” *Software Quality Journal*, vol. 8, pp. 211–231, 2000.

[52] A. Brandon-Jones and N. Slack, *Quantitative Analysis in Operations Management*. Financial Times Prentice Hall, 2008, p. 176.

[53] B. Martens, M. Walterbusch, and F. Teuteberg, “Costing of Cloud Computing Services: A Total Cost of Ownership Approach,” *2012 45th Hawaii International Conference on System Sciences*, pp. 1563–1572, Jan. 2012.

[54] J. David, D. Schuff, and R. S. Louis, “Managing your total IT cost of ownership,” *Communications of the ACM*, vol. 45, no. 1, pp. 101–106, 2002.

[55] D. Drury, “Determining IT TCO: Lessons and extensions.,” *9th ECIS*, pp. 825–836, 2001.

[56] B. G. Ferrin and R. E. Plank, “Total cost of ownership models: An exploratory study,” *Journal of Supply chain management*, no. August, pp. 18–29, 2002.

[57] P. Turney, “Activity-based costing: An emerging foundation for performance management,” *Cost Management*, 2010.

[58] Institute of Management Accountants, “Strategic Cost Management: Implementing Activity Based Costing,” *North American Die Casting Association(USA),*. Institute of Management Accountants, 2006.

[59] R. Cooper and R. Kaplan, “Activity-based systems: Measuring the costs of resource usage,” *Accounting Horizons*, vol. 6, no. 3, pp. 1–13, 1992.

[60] Encyclopedia of Management, “Activity-Based Costing.” .

[61] R. Kazman, J. Asundi, and M. Klein, “Quantifying the costs and benefits of architectural decisions,” in *Proceedings of the 23rd international conference on Software engineering*, 2001, pp. 297–306.

[62] R. L. Nord, M. R. Barbacci, P. Clements, R. Kazman, M. Klein, L. O’Brien, and J. E. Tomayko, “Integrating the Architecture Tradeoff Analysis Method (ATAM) with the cost benefit analysis method (CBAM).” Carnegie Mellon University Software Engineering Institute, p. 35, 2003.

[63] M. M. Parker and R. J. Benson, “Enterprisewide information economics: latest concepts,” *Information Systems Management*, vol. 6, no. 4, pp. 7–13, 1989.

[64] R. Dillibabu and K. Krishnaiah, “Cost estimation of a software product using COCOMO II.2000 model – a case study,” *International Journal of Project Management*, vol. 23, no. 4, pp. 297–307, May 2005.

[65] B. Boehm, R. Valerdi, J. A. Lane, and W. A. Brown, “COCOMO Suite Methodology and Evolution,” *Cross Talk*, vol. 18, no. 4, pp. 20–25, 2005.

[66] B. Boehm, B. Clark, E. Horowitz, and C. Westland, “Cost models for future software life cycle processes: COCOMO 2.0,” *Annals of Software Engineering*, vol. 1, no. 1, pp. 57–94, 1995.

[67] R. C. Tausworthe, “The work breakdown structure in software project management,” *Journal of Systems and Software*, vol. 1, pp. 181–186, Jan. 1980.

[68] C. L. Desmond, “Project Management for Telecommunications Managers,” in *Project Management for Telecommunications Managers*, 2004, pp. 71–83.

[69] B. Golany and A. Shtub, “Work Breakdown Structure,” in *Handbook of Industrial Engineering: Technology and Operations Management*, Third., G. Salvendy, Ed. Wiley; 3rd edition, 2007, pp. 1263–1280.

[70] B. Boehm, C. Abts, and S. Chulani, “Software development cost estimation approaches – A survey,” *Annals of Software Engineering*, vol. 10, no. 1–4, pp. 177–205, 2000.

[71] D. Ortloff, T. Schmidt, K. Hahn, T. Bieniek, G. Janczyk, and R. Brück, “Details for Managing a Project,” in *MEMS Product Engineering: Handling the Diversity of an Emerging Technology. Best Practices for Cooperative Development*, Springer; 2014 edition, 2013, pp. 173–188.

[72] D. J. Bryde and R. Joby, “Product-based planning: the importance of project and project management deliverables in the management of clinical trials,” *R&D Management*, vol. 37, no. 4, pp. 363–377, Sep. 2007.

[73] A. Szychta, “Time-Driven Activity-Based Costing in Service Industries.,” *Social Sciences*, vol. 67, no. 1, pp. 49–60, 2010.

[74] H. Jonkers, M.-E. Iacob, M. M. Lankhorst, and P. Strating, “Integration and analysis of functional and non-functional aspects in model-driven e-service development,” in *Ninth IEEE International EDOC Enterprise Computing Conference (EDOC’05)*, 2005, pp. 229–238.

[75] ISO, “ISO/IEC TR 9126-2.” p. 86, 2002.

[76] B. Mutschler, J. Bumiller, and M. U. Reichert, “An Approach to quantify the Costs of Business Process Intelligence.,” 2005.

[77] A. F. A. Fadzlah, “Timeliness Measurement Model: A mathematical approach for measuring the timeliness of handheld application usage,” *International Journal of New Computer Architectures and their Applications (IJNCAA)*, vol. 2, no. 3, pp. 431–450, 2012.

[78] C. Potts, “Using Structural Performance Ratios to Guide Investments in Enterprise Architecture,” *Journal of Enterprise Architecture*, vol. 6, no. 4, pp. 14–18, 2010.

[79] A. Abediniyan, M. Mohsenzadeh, and M. A. Dezfouli, “A new Approach towards Cost and Benefit Enterprise Architecture Analysis,” *International Journal of Computer Applications Technology and Research*, vol. 2, no. 2, pp. 160–165, 2013.

[80] V. R. Basili, G. Caldiera, and H. D. Rombach, “The goal question metric approach,” *Encyclopedia of software engineering*, vol. 2, pp. 528–532, 1994.

[81] R. E. Park, W. B. Goethert, and W. A. Florac, “Goal-Driven Software Measurement — A Guidebook,” no. August. p. 189, 1996.

[82] V. R. Basili, J. Heidrich, M. Lindvall, J. Munch, M. Regardie, D. Rombach, C. Seaman, and A. Trendowicz, “GQM+ Strategies: A comprehensive methodology for aligning business strategies with software measurement,” in *Tagungsbad des DASMA-Software-Metrik-Kongresses (MetriKon 2007)*, 2007, no. MetriKon.

[83] S. Buckl, F. Matthes, and C. M. Schweda, “A Technique for Annotating EA Information Models with Goals,” pp. 113–127, 2010.

[84] W. Goethert and J. Siviy, “Applications of the Indicator Template for Measurement,” no. September. p. 47, 2004.

[85] V. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Munch, D. Rombach, and A. Trendowicz, “Bridging the Gap between Business Strategy and software Development.” pp. 1–16, 2007.

[86] V. Basili, J. Heidrich, M. Lindvall, J. Münch, C. Seaman, M. Regardie, and A. Trendowicz, “Determining the impact of Business strategies using pronciples from goal-oriented measurement,” in *Proceedings of the 9 Internationale Tagung Wirtschaftsinformatik (WI2009)*, 2009, vol. 1, no. Wi, pp. 545–554.

[87] V. Mandić, V. Basili, L. Harjumaa, M. Oivo, and J. Markkula, “Utilizing GQM+ Strategies for business value analysis: An approach for evaluating business goals,” in *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 2010, pp. 20–29.

[88] J. Munch, F. Fagerholm, P. Kettunen, M. Pagels, and J. Partanen, “Experiences and Insights from Applying GQM+Strategies in a Systems Product Development Organisation,” *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 70–77, Sep. 2013.

[89] K. Petersen, C. Gencel, N. Asghari, and S. Betz, “An elicitation instrument for operationalising GQM+Strategies (GQM+S-EI),” *Empirical Software Engineering*, pp. 1–38, Mar. 2014.

[90] T. Tahir and C. Gencel, “A structured goal based measurement framework enabling traceability and prioritization,” *2010 6th International Conference on Emerging Technologies (ICET)*, pp. 282–286, Oct. 2010.

[91] A. M. Bhatti, H. M. Abdullah, and C. Gencel, “A Model for Selecting an Optimum Set of Measures in Software Organizations,” in *Software Process Improvement*, vol. 42, R. V. O’Connor, N. Baddoo, J. Cuadrago Gallego, R. Rejas Muslera, K. Smolander, and R. Messnarz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 44–56.

[92] F. García, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini, and M. Genero, “Towards a consistent terminology for software measurement,” *Information and Software Technology*, vol. 48, no. 8, pp. 631–644, Aug. 2006.

[93] M. A. Babar and I. Gorton, “Comparison of Scenario-Based Software Architecture Evaluation Methods,” in *11th Asia-Pacific Software Engineering Conference*, 2004, pp. 600–607.

[94] R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere, “The architecture tradeoff analysis method,” in *Engineering of Complex Computer Systems, 1998. ICECCS’98. Proceedings. Fourth IEEE International Conference*, 1998, pp. 68–78.

[95] R. Kazman, H. P. In, and H.-M. Chen, “From requirements negotiation to software architecture decisions,” *Information and Software Technology*, vol. 47, no. 8, pp. 511–520, Jun. 2005.

[96] A. Teka, N. Condori-Fernández, I. Kurtev, D. Quartel, and W. Engelsman, “Change Impact Analysis of Indirect Goal Relations: Comparison of NFR and TROPOS Approaches Based on Industrial Case Study,” in *Model-Driven Requirements Engineering Workshop (MoDRE), 2012 IEEE*, 2012, pp. 58–67.

[97] J. Wang, J. Li, Q. Wang, H. Zhang, and H. Wang, “A simulation approach for impact analysis of requirement volatility considering dependency change,” in *Requirements Engineering: Foundation for Software Quality*, Springer Berlin Heidelberg, 2012, pp. 59–76.

[98] A. Teka, “Analysis of Indirect Influence Relations in Goal-Oriented Requirements Engineering Thesis report for the degree of Master of Science in,” University of Twente, 2012.

[99] R. Ellis-Braithwaite, R. Lock, R. Dawson, and B. Haque, “Towards an approach for analysing the strategic alignment of software requirements using quantified goal graphs,” 2013.

[100] H. Kaiya, H. Horai, and M. Saeki, “AGORA: attributed goal-oriented requirements analysis method,” *Proceedings IEEE Joint International Conference on Requirements Engineering*, pp. 13–22, 2002.

[101] A. W. Schneider, C. Schulz, and F. Matthes, "Goals in Enterprise Architecture Management - Findings from Literature and Future Research Directions," *2013 IEEE 15th Conference on Business Informatics*, pp. 284–291, Jul. 2013.

[102] V. Popova and A. Sharpanskykh, "Modeling organizational performance indicators," *Information Systems*, vol. 35, no. 4, pp. 505–527, Jun. 2010.

[103] E. C. S. Cardoso, "Challenges in Performance Analysis in Enterprise Architectures," *2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*, pp. 327–336, Sep. 2013.

[104] F. Matthes, I. Monahov, A. W. Schneider, and C. Schulz, "Towards a Unified and Configurable Structure for EA Management KPIs," in *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation*, vol. 131, S. Aier, M. Ekstedt, F. Matthes, E. Proper, and J. L. Sanz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 284–299.

[105] F. Matthes, I. Monahov, A. Schneider, and C. Schulz, "EAM KPI Catalog v1.0," 2011. [Online]. Available: https://wwwmatthes.in.tum.de/pages/19kw70p0u5vwv/EAM-KPI-Catalog.

[106] F. Matthes, I. Monahov, A. Schneider, and C. Schulz, "A Template-Based Design Method to Define Organization-Specific KPIs for the Domain of Enterprise Architecture Management," in *DASMA Software Metrik Kongress*, 2012, vol. 19.

[107] A. Del-Río-Ortega, "On the Definition and Analysis of Process Performance Indicators," University of Sevilla, 2012.

[108] A. Del-Río-Ortega, M. Resinas, C. Cabanillas, and A. Ruiz-Cortés, "On the definition and design-time analysis of process performance indicators," *Information Systems*, vol. 38, no. 4, pp. 470–490, Jun. 2013.

[109] L. F. C. Rojas and C. M. Z. Jaramillo, "Executable pre-conceptual schemas for representing key performance indicators," in *2013 8th Computing Colombian Conference (8CCC)*, 2013, pp. 1–6.

[110] V. Popova and A. Sharpanskykh, "Formal modelling of organisational goals based on performance indicators," *Data & Knowledge Engineering*, vol. 70, no. 4, pp. 335–364, Apr. 2011.

[111] U. Frank, D. Heise, H. Kattenstroth, and H. Schauer, "Designing and Utilising Business Indicator Systems within Enterprise Models – Outline of a Method," in *Proceedings of Modeling Business Information Systems Conference (MobIS 2008)*, 2008, pp. 89–1.

[112] D. Barone, L. Jiang, D. Amyot, and J. Mylopoulos, "Reasoning with Key Performance Indicators," in *The Practice of Enterprise Modeling*, vol. 92, P. Johannesson, J. Krogstie, and A. L. Opdahl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 82–96.

[113] A. Maté, J. Trujillo, and J. Mylopoulos, "Conceptualizing and specifying key performance indicators in business strategy models," in *Proceeding CASCON '12, Conference of the Center for Advanced Studies on Collaborative Research*, 2012, pp. 102–115.

[114] A. Pourshahid, G. Richards, and D. Amyot, "Toward a Goal-Oriented, Business Intelligence Decision-Making Framework," in *E-Technologies: Transformation in a Connected World*, vol. 78, G. Babin, K. Stanoevska-Slabeva, and P. Kropf, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 100–115.

[115] A. A. Priyanto, "Bringing enterprise architecture to the boardroom." 22-Nov-2013.

[116] M. E. Iacob, L. O. Meertens, H. Jonkers, D. a. C. Quartel, L. J. M. Nieuwenhuis, and M. J. Sinderen, "From enterprise architecture to business models and back," *Software & Systems Modeling*, Dec. 2012.

[117] M. R. Davoudi, F. S. Aliee, and M. Mohsenzadeh, "A New Approach towards Enterprise Architecture Analysis," in *International Conference on Enterprise Information Systems and Web Technologies (EISWT-09)*, 2009, pp. 44–51.

[118] K. Wiegers, "First things first: prioritizing requirements," *Software Development*, vol. 7, no. 9, pp. 48–53, 1999.

[119] P. Berander and A. Andrews, "Requirements Prioritization," in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Eds. Berlin/Heidelberg: Springer-Verlag, 2005, pp. 69–94.

[120] H. Jonkers, I. Band, and D. Quartel, *ArchiSurance Case Study*. 2012.

[121] C. M. Staff, "Retention Metrics You Should Be Using," 2009. [Online]. Available: http://www.chiefmarketer.com/news/retention-metrics-you-should-be-using-04082009.

[122] Klipfolio, "Customer Retention," 2013. [Online]. Available: http://www.klipfolio.com/resources/kpi-examples/retail/customer-retention.

[123] P. Drucker, "How To Calculate Customer Retention." [Online]. Available: http://www.evergage.com/blog/how-calculate-customer-retention/.

[124] C. Basu, "The Difference Between Revenue Retention and Revenue Growth." [Online]. Available: http://smallbusiness.chron.com/difference-between-revenue-retention-revenue-growth-34421.html.

[125] K. Doumi, S. Baïna, and K. Baïna, "Modeling Approach Using Goal Modeling and Enterprise Architecture for Business IT Alignment," in *Model and Data Engineering*, Springer Berlin Heidelberg, 2011, pp. 249–261.

[126] A. MacLean, R. M. Young, V. M. E. Bellotti, and T. P. Moran, "Questions, options, and criteria: Elements of design space analysis," *Human–computer Interaction*, vol. 6, pp. 201–250, 1991.

[127] M. A. Babar, I. Gorton, and B. Kitchenham, "A Framework for Supporting Architecture Knowledge and Rationale Management," in *Rationale Management in Software Engineering*, A. H. Dutoit, R. McCall, I. Mistrík, and B. Paech, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 237–254.

[128] A. H. Dutoit, T. Wolf, B. Paech, L. Borner, and J. Ruckert, "Using Rationale for Software Engineering Education," in *18th Conference on Software Engineering Education & Training (CSEET'05)*, 2005, pp. 129–136.

[129] O. Zimmermann, T. Gschwind, J. Küster, F. Leymann, and N. Schuster, "Reusable Architectural Decision Models for Enterprise Application Development," in *Software Architectures, Components, and Applications*, vol. 4880, S. Overhage, C. A. Szyperski, R. Reussner, and J. A. Stafford, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 15–32.

[130] O. Zimmermann, J. Koehler, and F. Leymann, "Architectural decision models as micro-methodology for service-oriented analysis and design," in *SEMSOA Workshop*, 2007.

[131] D. Rchid, O. Bouksour, and Z. Beidouri, "The Activity Based Costing method opportunity to assess and master the aircraft maintenance service cost for Third Party : a case study," *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 1, pp. 699–706, 2013.

APPENDIX I – Benefits categories and indicators

Operational benefits					
	Process	**Customer**	**Finance**	**Innovation**	**HR**
High level benefit	Improve process efficiency	Meet customer needs more efficient	Reduce cost	Increase productivity	
Indicator benefits	- Error/rework reduction - Faster processing - Consistent data - Reduction in processing time - Increased throughput[1] - Productivity improvement - Quality improvement [2] - Centralization of adm. Activities [3] - Savings through new work approaches [4] - Faster order response [5]	- Improved response time - Reduced customer complaints - Reduced errors [1] - Customer services improvement [2]	- Reduced inventory carrying cost - Lower labor cost [1] - Decrease cost of circulation - Increase the turnover rate of capital [5]	- Power user involvement in user training for operational tasks [1] - Savings through new work approaches [4]	
Tactical/Managerial benefits					
High level benefit	Improve tactical decision making	Identify and meet customer needs proactively	Increase revenues	Make workers more effective decision makers	Worker empowerment
Indicator benefits	- Improved work scheduling - Improved work assignment - Improved access to information - Improved quality management - Improved control [1] - Better resource management - Improved decision-making and planning - Performance improvement [2] - Improve coordination of the circuit of capital [5]	- Better customer expectation setting - Improved customer satisfaction - Improved engine repair scheduling and delivery [1]	- Better forecasting - Increase market Share [1] - Revenue enhancement benefits [4]	- Training for access of enterprise information - Training for decision making skills [1]	- Worker empowerment for taking actions [1]
Strategic benefits					
High level benefit	Adapt to radical environment changes routinely	Meet new customers needs or new needs of customers	Improve market value	Absorb radical change routinely	
Indicator benefits	- Technology changes - Regulatory changes - Competition changes [1] - Support business alliance - Build external linkages [2]	- Increased customer base - Partnership with customers [1] - Generate product differentiation [2]	- Growth capitalization - New markets [1] - Built cost leadership [2] - Compete on product markets [5]	- Change management Processes - Breadth and broader horizon [1] - *Build business innovation* [2]	
IT-Infrastructure benefits					
Indicator benefits	- Increased IT-infrastructure capability [2]		- IT cost reduction [2] - Reduction in IS maintenance cost [3]	- *Build business flexibility for current and future changes* [2]	
Organizational benefits					
Indicator benefits				- *Support org. changes* - *Facilitate Business learning* - *Empowerment* [2]	- *Built common visions*[2]

Figure 113 - Benefits categories and indicators, obtained from Eckartz [50]

APPENDIX II – Cost Categories, Cost factors and EA layers

A variety of cost categories regarding IT investments and projects have been identified in literature and in practice. The cost categories presented in Table 75 comprise a re-categorization based on the categories used in Life Cycle Costing and in Total Cost of Ownership ([48] [45] [54] [55]). Additionally, for every cost category or stage of the life cycle of an IT project, a number of cost factors have also been found in literature regarding IT investments and costs considered in TCO, both easily addressed and hidden costs ([48] [54] [55] [42] [45]). The cost factors have been grouped from two perspectives. The first classification (columns) is based on the most common costs related to IT. These categories (hardware, software, network and communications, etc.) can be seen as high-level categories of cost factors. The second classification (colored rows) indicates in which EA layer each cost factor resides. This classification per EA layer has been made based on the architectural concepts found in each layer according to ArchiMate® Specification [17].

Table 75 can be used as a guideline for measuring costs in an EA model. The aggregated high-level categories of costs factors can serve as the input for the calculations performed in each layer, while the output can be the joint result of the calculations put in the cells of the table.

		Cost Factors					
Cost Categories		**Hardware Costs**	**Software costs**	**Network & Communications**	**Personnel costs (IT staff & Users)**	**Other costs**	**EA Layers**
	Acquisition & Implementation				- Pre-planning Commissioning/Consultancy costs - Initial training costs - Downtime costs (non-working employees/users) during install	- Initial site planning - Floor space acquisition/renovation - Compliance costs	**Business Layer**
			- Operating System purchase/license - Application purchase (one-time cost): - Front-end applications - Development/migration SW purchase - Security and		- SW installation - SW migration labor		**Application Layer**

			encryption SW - Systems management SW				
		- Purchase costs: * Front-end costs (monitors, printers, scanners, etc.) * Backend costs (server, PC client system, W/S client system, etc.)	- Operating System purchase/license	- Network/ Communications HW - Network/ Communications SW	- HW installation - SW installation - Installation of communication equipment - Initial Network setup - SW migration labor		**Infrastructure Layer**
Administr a-tion costs	**Operating costs**				- Administration labor - Continuing contract labor - Continuing training - User help & Support costs - Auditing - Futz factor	- Costs for measuring cost factors	**Business Layer**
			- Periodic SW license fees		- Troubleshooting - System management	- Security costs (disaster, recovery services, etc.)	**Application Layer**
		- HW lease expenses	- Periodic SW license fees	- Communication & Network charges (line usage, Internet service provider, etc.)	- Troubleshooting - System management	- Power consumption (Electricity for equipment and for air-conditioning) - Security costs (disaster, recovery services, etc.)	**Infrastructure Layer**
	Control costs				- (specialized) Training		**Business Layer**

			- Centralization/ standardization SW costs		- Installation costs - Migration labor costs		**Application Layer**
		- Centralization/ standardization HW costs			- Installation costs - Migration labor costs		**Infrastructure Layer**
	Maintenance costs				- Downtime costs (non-working employees/users) during maintenance		**Business Layer**
			- SW maintenance/ warranty fees - Downtime costs (failures and install/upgrade)		- Maintenance labor costs		**Application Layer**
		- HW maintenance fees	- HW maintenance fees - Downtime costs (failures and install/upgrade)		- Maintenance labor costs		**Infrastructure Layer**
	Ongoing Change & Growth costs	- Capacity planning (in-house and/or consultancy costs)	- Capacity planning (in-house and/or consultancy costs)	- Network change planning costs	- Capacity planning (in-house and/or consultancy costs) - Additional user training - Evaluation costs for new/upgraded versions of HW and SW - Downtime costs (non-working employees/users) during install/upgrade	- (additional) Site expansion, consolidation, preparation, renovation Compliance costs	**Business Layer**
			- Operating system upgrade - Migration SW purchases		- Operating system upgrade labor		**Application Layer**

		- Additional systems (servers, client systems) - System upgrades - Storage space expansion	- Operating system upgrade - Migration SW purchases	- Additional network/communication HW and SW	- HW reconfiguring/setup - Operating system upgrade labor - Network changes - Administration costs		**Infrastructure Layer**
	Disposal costs						**Business Layer**
			- Cost of demolition, scrapping or selling the SW application				**Application Layer**
		- Cost of demolition, scrapping or selling HW component	- Cost of demolition, scrapping or selling the SW application	- Cost of demolition, scrapping or selling network/communication component			**Infrastructure Layer**

Table 75 - Cost Categories, Cost factors and EA layers

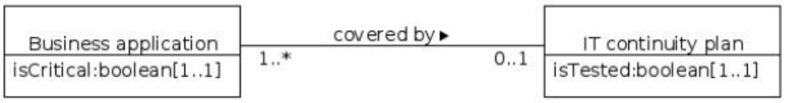
APPENDIX III – Example EAM KPI Structure

Application continuity plan availability

Description

A measure of how completely IT continuity plans for business critical applications have been drawn and tested up for the IT's application portfolio.

Information model



Organization-specific instantiation

Mapping:

Name in model	Mapped name	Contacts	Data sources
Business application	Application	J. Doe	EA repository
isCritical	criticality	J. Doe	EA repository
covered by	covers	R. Miles	Risk Mgmt. rep.
IT continuity plan	Disaster plan	R. Miles	Risk Mgmt. rep.
isTested	tested	R. Miles	Risk Mgmt. rep.

Properties:

KPI property	Property value	Best-practice
Measurement frequency	Yearly	Quarterly
Interpretation		Good > 80% Normal 60% - 80% Problematic < 60%
KPI consumer	J. Smith	
KPI owner	J. Doe	
Target value	100% in 2015	80%
Planned value(s)	25% in 2013 75% in 2014	70%, 75%
Tolerance value(s)	5%	
Escalation rule	n.a.	

Goals

Ensure compliance
Foster innovation
Improve capability provision
Improve project execution
Increase disaster tolerance
Increase homogeneity
Increase management satisfaction
Increase transparency
Reduce operating cost
Reduce security breaches

Calculation

The number of critical applications where tested IT continuity plan available divided by the total number of critical applications.

Code

EAM-KPI-0001

Sources

CobiT 4.0

Layers

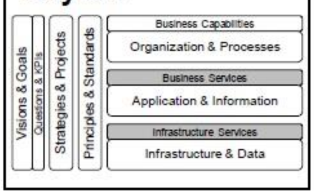


Figure 114 - Example EAM KPI Structure

APPENDIX IV – Additional Goal Analysis Methods

1. Integration of Business Value Analysis and GQM+ Strategies

An attempt is presented by Mandić et al. [87] for combining Business Value Analysis (BVA) with GQM+ Strategies. The authors introduce the concept of a 'value goal' in order to analyze the value aspect of business goals. A BVA approach takes into account cost and benefit estimates, time constraints and risks. These factors are implemented through value goals by exploiting the alignment and integration among goals and strategies in different levels provided by GQM+ Strategies. The value analysis in the proposed method is performed in a parallel with the grid derivation process.

Value goals follow the template used in the GQM+ Strategies approach for business and lower-level goals. The only difference though is that the elements 'activity', 'focus', 'object' and 'relations' are pre-set (static) to 'evaluate', 'value', 'business' and 'top-level business goals' respectively, since the value goals serve a specific purpose to assess the value of the corresponding business goals and strategy elements of the grid. Moreover, value goals aim to propagate the rationale of investment-related decisions to all levels as well as to integrate estimations regarding costs and benefits from all levels.

Furthermore, there are two types of value goals, i.e. the top-level value goals and the lower-level ones. Their difference relies on the fact that "the top-level value goal defines the model of acceptable risk for the available size of investment, and based on context information, specifies the time-period of analysis" [87]. The refinement of value goals is based on the evaluation of the value of the business goals of a specific level: "A value goal estimates cost and benefits of the corresponding business goal for the specified time period, and uses cost-benefit information to assess acceptable risk with the model specified by the top-level value goal" [87].

A set of generalized equations are also proposed by the authors for measuring the costs and benefits as well as estimating the acceptable risk and assessing the risk level of the business goals against it. The estimations, though, are qualitative. Additionally, the timeframe of an investment and the different timeframes of the business goals are taken into account in the calculations. The added value of the integration of the two methods is the insight regarding the quality of the GQM+ Strategies as critical sub-grids can be identified due to inserting the risk levels.

2. Strategic Alignment as a Goal Analysis Aspect

2.1. Goal Modeling and EA for IT Alignment

The purpose of the method proposed by Doumi et al. [125] is to incorporate the strategic alignment notion in the modeling of the enterprise strategy. Strategic alignment comprises the overall coherence of decisions and actions made by the organization with the strategic objectives of the organization. Information systems are viewed as the means for accomplishing the alignment, while enterprise architecture as a supportive concept for analyzing the alignment of the information systems. The focus of the paper by Doumi et al. [125] is to enhance the business-IT alignment through a method for representing and evaluating it.

'Alignment' in the approach is considered as the set of links between the model representing the strategy (i.e. the goal model) and the IS model (i.e. the EA model). "[T]he degree of alignment is measured by comparing (i) the set of linkages between elements of the IS model and elements of strategic model and (ii) the aggregate maximum possible links between these models" [125].

The element of 'Target' is introduced in the alignment approach. A 'target' incorporates the notion of the indicator which expresses the accomplishment of a goal or objective at the strategic level, i.e. the goal level. Additionally, it is the concept that connects the strategic level with the functional level, since at the former the target is defined while at the latter is executed. A matrix is constructed for showing the realization of an indicator by parts of the functional architecture grouped based on the business areas they belong to. A misalignment or lack of coverage of an indicator can then be discovered. The notions of neighborhoods and building blocks are used for reorganizing the information systems architecture in their related business areas, but basically what the authors want to achieve is to directly assign the indicator (which is derived by a target) to a specific element of the information systems architecture and to the process the element contributes to.

To conclude, the alignment approach of the strategic model and the information systems model is achieved through the matrix which indicates the existence of a relationship between an indicator or a task (belonging in the strategic model) and an element of the information systems model. By using the links between the goals and the tasks, the correspondence of the IT systems to the goals or tasks can be derived.

2.2. Strategic Alignment Analysis of Software Requirements using Quantified Goal Graphs

The strategic alignment analysis approach, described by Ellis-Braithwaite et al. [99], focuses on providing justification for the contribution of software requirements to the strategy through goal graphs. The authors attempt to extend the contribution relationship in a goal graph by combining it with the uncertainty, confidence, non-linear causation and utility concepts for improving the demonstration of the strategic alignment of the requirements.

The *strategic alignment of a requirement* deals with the demonstration of the benefits that the requirement offers to the organization. Determining how the requirement contributes to the benefits, as defined by the stakeholders, solves any uncertainties regarding the need of the requirement. The strategic alignment analysis approach should be applied "after the high-level requirements have been elicited, so that resources are not wasted eliciting lower level requirements that do not align well to business strategy" [99]. This approach is a quantitative approach because of the quantitative nature of requirements which promotes quantitative analysis for proving their contribution to the strategy.

For the construction of the goal graph, the approach is based on the GQM$^+$ Strategies formalization template for goals. A few changes are proposed for better alignment of the textual description of a goal or objective with the visual representation of the graph. The satisfaction attribute of a goal is still defined by its magnitude, while for a requirement a metric is specified as its satisfaction criterion.

Single-point and *multi-point* goal graph quantifications are alternatives described for determining the predictions of quantified link contributions. In the former, the full satisfaction of the objective or the requirement is taken into account, while the latter considers the effect of the partial satisfaction of an objective to the objectives related to it as well as to the whole chain of goals and it is represented through a *table function* together with a chosen interpolation method. The predictions are based on the target levels of satisfaction of either the requirements or the goals.

Ellis-Braithwaite et al. distinguish between 'confidence' which refers to the belief that a predicted value is correct and 'uncertainty' which refers to "possible values for the unknown quantity" [99]. Modeling the confidence indicates the existence of the risk for realizing the strategic alignment. A discussion is also provided in the paper for relating the concept of utility with the level of satisfaction of a goal.

The importance of measuring the current value of a magnitude attribute of an objective is also stressed. Acquiring the current value can be the start for recording values over time in order to evaluate the improvement or change of the system and provide a comparison basis for the future. Additionally, documentation of the assumptions made during defining the contribution links and scores is considered crucial. Finally, a tool has been developed for supporting the approach by generating the diagrams automatically based on information inputs regarding requirements, objectives and contribution data.

The basic contribution of this approach regards the goal contribution links which support the linkage between business value and requirements. The contribution links can be expressed and represented in different ways depending on the degree of satisfaction of the relative elements and they can be viewed as "quantitative causation relationships (…) through more than one level of goal abstraction, in order to understand the effects of partial requirement satisfaction on high-level goals" [99].

3. Requirements Change Impact Analysis Techniques

3.1. NFR based Fuzzy Logic Approach for evaluating contribution links

Teka in his thesis [98] has examined two approaches for performing change impact analysis on indirect contribution relationships of goals and requirements. The two approaches compared in his thesis are TROPOS and NFR based Fuzzy Logic. The selection of TROPOS and NFR as the candidates for the comparison relied on the fact that they are easy to understand and use as well as well-documented methods [96]. Since TROPOS approach is considered more appropriate for *qualitative* reasoning of high-level goal analysis, it will not be further described in this section. The combination of the NFR approach with fuzzy logic reasoning transforms NFR reasoning from qualitative to *quantitative*, which makes it a valuable approach for analyzing satisfaction and contribution levels of goals and relationships [96] in a more concrete way.

While NFR itself provides a method for qualitative reasoning resulting in vague conclusions which limit the value of the method, "fuzzy logic based on fuzzy sets (…) uses a concept of membership function to determine the membership value of a certain input to a set" [96]. The sets have a trapezoidal form (Figure 115) which results in the assignment of up to two fuzzy sets to a given value [96].

Teka's methodology employs the six satisfaction levels of goals and seven types of contribution relationships provided by the NFR framework (from which only five are used in the proposed approach [96]), and maps them to fuzzy sets (i.e. specific intervals of the -100 to 100 scale are assigned to each level and type, which can be found in [96]). The satisfaction levels and the relation types are illustrated in Figure 116. The analysis consists of the two following phases:

1. **First phase**. Firstly, the satisfaction levels for performing the analysis are assigned to the leaf goals of the graph in collaboration with the stakeholders as input values [96], while the "individual impact of an offspring contribution towards a parent for each contribution relationship" needs to be defined initially as well [98].
   Additionally, rules for propagating the satisfaction levels of the goals across AND/OR goal decomposition relationships are provided for assigning satisfaction levels in every goal (Figure 116).
   Then, the initial goal satisfaction levels and contribution strengths should be assigned a membership value based on the goal satisfaction and contribution relations (already specified) fuzzy sets. Linear equations are used for defining the membership values. Examples are shown in both [96] and [98].
   In the next step, the membership values of the satisfaction levels and influence relation strength are combined, resulting probably in two values. Rules are provided for combining the pairs [96][98]. These two values (membership values) are re-assigned to the fuzzy sets resulting in the grey shapes depicted in Figure 115. The vertical axis in the figure represents the membership values, while the horizontal the satisfaction levels.
2. **Second phase.** The two results (membership values) of the previous phase have to be aggregated and the corresponding value needs to be changed into an aggregated value for the satisfaction level of the goal. The technique used for calculating this value is finding the centroid of the shaded region in Figure 115, which will present the satisfaction level of the goal. When the centroid is found, an opposite process (de-fuzzification) is followed for assigning the corresponding crisp satisfaction level to the goal under analysis.

For implementing the two phases, algorithms have been provided by Teka in his thesis [98]. While this approach enables the assignment of numerical values during the goal contribution analysis, there are a few limitations observed. The most important one entails the final assignment of a zero satisfiability value in case of conflicting goal influences. A zero satisfiability value, though, can also be assigned to a goal which is neither satisfied nor denied. Thus, ambiguity can be created.

Among the benefits of the NFR based Fuzzy Logic approach are its ability to combine two or more influence effects and to provide concrete values; hence, enabling more detailed goal analysis and a deeper investigation of goals and requirements.

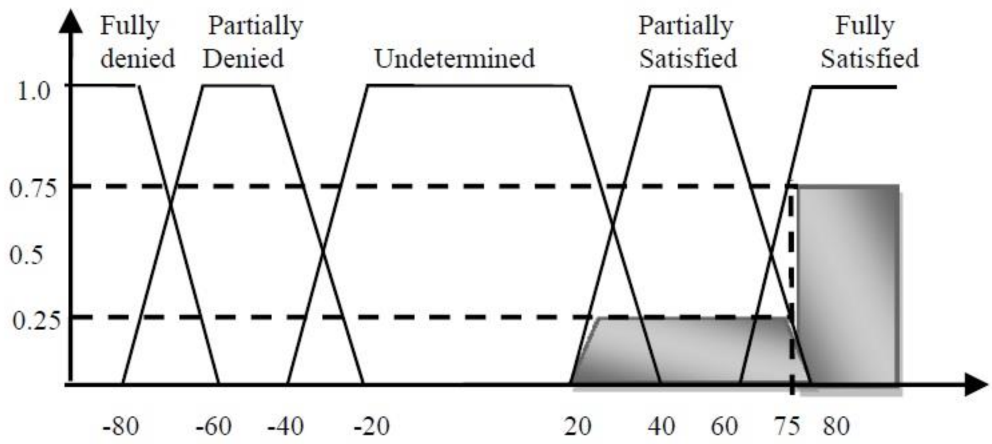


Figure 115 - Example of Fuzzy sets for Satisfaction levels, obtained from Teka et al. [96]

	Upon Parent label, given off-spring parent contribution types				
	Break(--)	*HUR T (-)*	*Unkn*	*HELP (+)*	*Make (++)*
Fully Denied (FD)	FS	PS	U	PD	FD
Partially Denied (PD)	PS	PS	U	PD	PD
Unknown (U)	U	U	U	U	U
Partially Satisfied (PS)	PD	PD	U	PS	PS
Fully Satisfied (FS)	FD	PD	U	PS	FS
Conflicting (C)	C	C	C	C	C

Figure 116 - Impact Analysis Rules for the NFR Framework, obtained from Teka et al. [96]

Integration of NFR based Fuzzy Logic in Architect tool

What is also worth to mention regarding the quantitative reasoning approach proposed by Teka [98], is the fact that the NFR based Fuzzy Logic approach (as well as the TROPOS qualitative reasoning approach) was combined with goal graphs developed by using the ArchiMate® motivation extension. Additionally, a proposal is made on how to integrate the goal contribution analysis approaches in the Architect tool. The designed integration of the approach in Architect resembles the DepRVSim simulation approach [97], since it borrows the concept of 'Change Event Generator' from DepRVSim, while the NFR based Fuzzy Logic approach is considered as part of the 'Quantitative/Qualitative Reasoning Engine' which handles the changes in the goal model in collaboration with the 'Influence Reasoning Rule Base' [98].

Finally, an additional algorithm is developed, which uses as a starting point the NFR based Fuzzy Logic (quantitative) approach, for dealing with goal graphs containing influence feedback loops [98]. Positive and negative feedback loops are observed and the corresponding behavior of the change effects on the graphs is described. The propagation algorithm reasons and computes the change percentage of the satisfaction level of the influencing goal and updates the influenced goals' satisfaction levels. The changes are based on the change percentage and the initial satisfaction values. A test case is employed for simulating the algorithm and demonstrating its usefulness.

3.2. DepRVSim – A simulation approach for requirements volatility and dependency relationships

DepRVSim, which stands for Requirement Volatility Simulation considering Dependency relationship, is a simulation approach for modeling the dependency and traceability relationships among requirements and the changes that occur in the structural dependency relationships [97]. The requirement changes, then, can motivate schedule or effort deviations in projects when linkages are provided between requirements and specific tasks of the project under development. For doing so, requirement change events are considered regarding: (i) *Requirement change type*: addition, modification or deletion, (ii) *Requirement change time*: when the requirement change event occurs, and (iii) *Degree of requirement modification* (if the requirement is modified).

According to Wang et al. [97], discrete-event simulation as the one of the DepRVSim approach "allows more detailed descriptions of activity, resource and work product and [is] more suitable for building fine-grained software process simulation models". Through DepRVSim, effects of requirement changes can be monitored and mapped on other requirements and (indirectly) on architecture components as well as on project plan deviations regarding effort and time. The structure of the DepRVSim approach is illustrated in Figure 117. While DepRVSim may provide a good basis for making estimations of the impact of requirements changes on a project plan, it does not define a specific methodology for developing a well-structured graph of goals and requirements and does not fully support the analysis of indirect influence relations among goals and requirements.

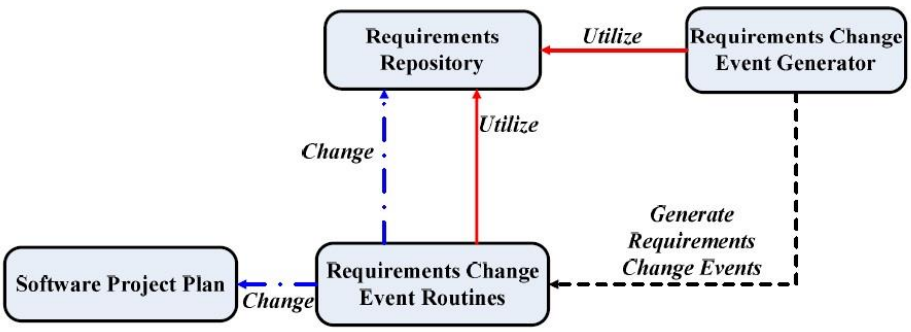


Figure 117 - DepRVSim Structure, obtained from Wang et al. [97]

3.3. AGORA – Attributed Goal-Oriented Requirements Analysis Method

AGORA is a method developed by Kaiya et al. [100] for supporting the existing goal-oriented analysis techniques by adding attributes, such as contribution and stakeholder's preference values, on the goal graphs and, hence, improving the goal and requirements refinement and decomposition tasks. The main advantage of the AGORA method is that it can be applied on top of or in parallel with existing methods as its main activity is the attachment of the additional attributes on the goal-graphs under development. AGORA is a quantitative analysis technique which can help in:

- Choosing and adopting a goal out of possible goal alternatives

- Recognizing conflicts among goals and among stakeholders' views on a goal
- Analyzing the impact of requirements changes
- Estimating the quality of requirements specifications by utilizing a number of artifacts (e.g. paths, contribution values, goal decomposition throughout the levels) becoming available during the elicitation process. A group of measures are defined in the paper by Kaiya et al. [100] for operationalizing the quality estimates (e.g. correctness, completeness, traceability).

The extensions to the goal-graphs provided by AGORA are the following:

1. **Attachment of *contribution values* on the edges (links) between goals.** This value is a number between -10 and 10. A contribution value denotes "the degree of the contribution of the goal to the achievement of its connected parent goal" [100].
2. **Attachment of *preference matrices* on goal nodes.** A preference matrix denotes the preference degree of each stakeholder for the goal attached to and his/her estimations for the preferences of the other stakeholders, as it shown in the example below:

	C	A	D
Customer (C)	8	-7	0
Administrator (A)	10	10	-10
Developer (D)	5	-10	0

   The diagonal numbers represent the preference of each stakeholder. The values are between -10 and 10, as for the contribution values. The stakeholders can fill in the matrix based on their own judgment but more systematic techniques such as AHP can be used for pair-wise comparison of the values in the matrix.
   The values in the matrix are used for assessing the gap of understanding of a goal among different stakeholders. Thus, they are indicators of potential conflicts among stakeholders' preferences.
3. **Attachment of *rationale statements* on attributes, nodes or edges.** Rationale statements provide explanation on decisions made while developing the graph. An example of their use is the justification of the preference values.

The process for creating the AGORA goal graph includes the following steps:

1. **Establishment of initial goals as customers' or organization's needs.** These goals are the roots of the end graph.
2. **Decomposition and refinement of goals into sub-goals.** In this step, the attachment of contribution values to the links and preference matrices to the nodes (goals) takes place, while sub-goals are related to their parent goals. Rationale statements are also filled in for providing clarity to the decisions made. When the variance in a preference matrix is high, the rationales will be analyzed for identifying the source of the conflict.
3. **Choosing and adopting goals from the alternatives.** This step is responsible for selecting suitable sub-goals based on the defined contribution values. A higher contribution value indicates a higher usefulness of this sub-goal for the parent goal comparing to other possible alternatives. Preference values are also utilized in a similar way. Depending on the view from which the goal graph is developed (e.g.

customer's view, manager's view or developer's view), the corresponding stakeholder's preferences can be valued more than others for selecting a sub-goal.

4. **Detecting and resolving conflicts on goals.** A conflict exists when :
    - the contribution value on the edge connecting two goals in negative (goal-to-goal conflict)
    - the diagonal elements of a goal's preference matrix have a large variance or are much deviated from their average value (stakeholders' conflict on a goal)

    Solutions are proposed by the authors for overcoming these conflicts, such as further decomposition of the sub-goal or stakeholder negotiation for resolving the issue.

The AGORA method can be terminated when the goals have become more concrete and operational. In terms of the ArchiMate® language, when they have turned into requirements.

The measures proposed in the paper by Kaiya et al. [100] are out of scope of this thesis, since they focus on the quality assessment of the sub-goals and, thus, of the specification process of the operational goals.

What is important, though, is the notion of attaching a variety of attributes on the edges of a goal graph and especially contribution values which provide justification for the refinement of goals and the selection of the most suitable concrete goals to guide the requirements' specification. The matrices can also be useful for prioritization decisions as well as for representing measures related to a node and their corresponding values or even relations between questions and measures in case of using a GQM-based method. Then, the documentation that accompanies the GQM-based techniques can be reduced and directly attached on the graph. A proper representation and visualization would be essential for keeping the graph readable and easy-to-understand.

4. A Design Rationale Approach - Questions, Options and Criteria

'Design rationale', according to MacLean et al. [126], is a representation for documenting the reasoning and argumentation regarding a specific artifact. Design rationale researchers have developed various methods and tools for recording design decisions, which usually include a small variance of node types and link types for structuring in a hierarchical way the questions which address the design issues [127]. Dutoit et al. [128] refer to such models as 'issue models' which represent individual decision-making elements as separate nodes and the relationships among them as links. Additionally, information such as the alternatives, their rationales as well as choices made along the design of an artifact can be also mapped on the questions of the model, providing justification for the discussion paths and the final outcome [127].

MacLean et al. [126] developed such a methodology – the Questions, Options and Criteria (QOC) approach – which aims to create a structured representation of the design alternatives and record the choices taken among the alternatives and decisions made for resulting in the final design. The three main elements of the approach, as denoted by its name, are:

- **Questions**, which aim to refine key issues leading to the alternatives by depicting the local contexts in more detail [126].
- **Options**, which are possible alternative solutions for the issues in stake and which provide answers to the Questions [126] [128].
- **Criteria**, which are bases for evaluating and selecting among the Options [126]. They are the desirable properties of the artifact [126] which can be non-functional

requirements, system design goals, test criteria [128] as well as non-technical factors, such as costs and available skills [129].

QOC diagrams (Figure 118) pose a design question, which leads to a set of options assessed by criteria. A selection of an option can generate new questions for further elaboration of the decision [126] [130] [129]. An 'issue', as defined by Dutoit et al. [128], represents the need to be solved by the application of the QOC approach. Thus, the issue is the starting point for developing the diagram and it can comprise a design issue, a possible defect, etc. Assessments also play an important role in the QOC approach, since they represent the evaluation of an option against a specific criterion [128]. The evaluation can be positive or negative, but its purpose is not to impose a decision or choice [126]. Instead, assessment's goal is to support the trade-off analysis.

QOC approach is an example of an argumentation-based method which enables rational design decisions through structured diagrams. This representation supports reflection on design decisions, reviewing decisions of other stakeholders as well as communication of decisions among various stakeholders in different stages of the design process (e.g. members of the current design team, developers, future maintenance designers), alternatives consideration, identification of relevant criteria and management of complex designs, since documentation of both requirements and constraints is possible [126] [128]. Furthermore, there are tools, such as Sysiphus (to which Dutoit et al. [128] refer), which provide functionality for linking relevant system model elements to the rationale models.

The linkage between the two types of models is also valuable for this thesis. Since criteria comprise, for example, non-functional requirements or characteristics of the system, they can be represented by the requirements (leaves) of the goal-graph, while issues can be derived by the high-level goals which drive the design decisions. Thus, QOC provides a methodology for generating design alternatives for the future architecture and an indirect linkage to both the goal model and the architecture model. The notion of relating criteria to goals and requirements aligns with the concept of 'bridging criteria' introduced by MacLean et al. [126] as well as with the properties of criteria provided by the authors.

A variety of other methods for representing the design rationale of a system exist. The selection of the QOC approach relies on the fact that it comprises a representative and suitable example method for creating design alternatives; these alternatives will be further assessed for defining the best to-be architecture for the migration based on the changes imposed by the goal level.

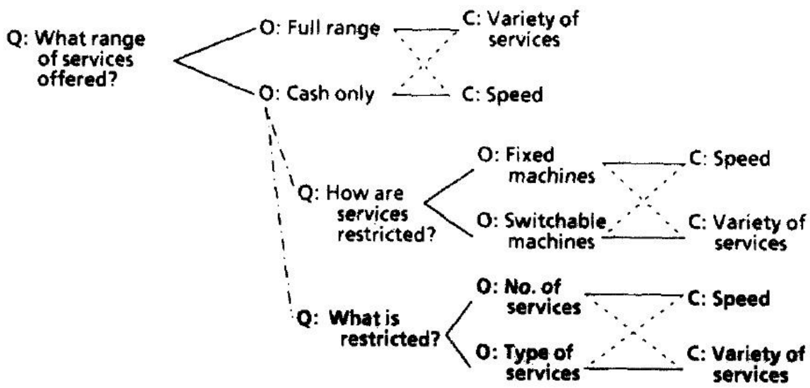


Figure 118 - QOC Diagram, obtained from [126]

APPENDIX V – EAGQ Model: Step Activities and Step Dependencies

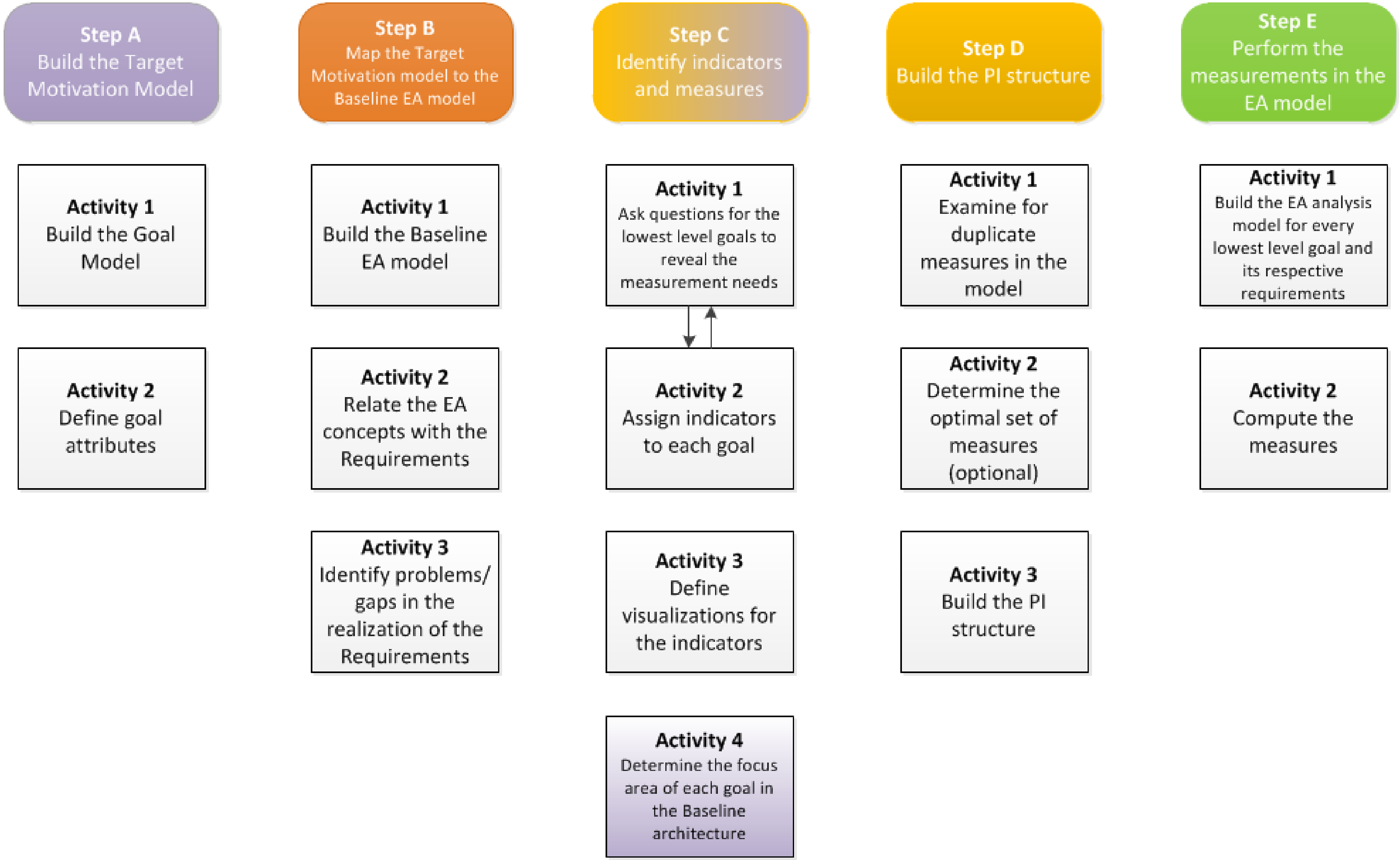


Figure 119 - EAGQ Method Steps A-E and their activities

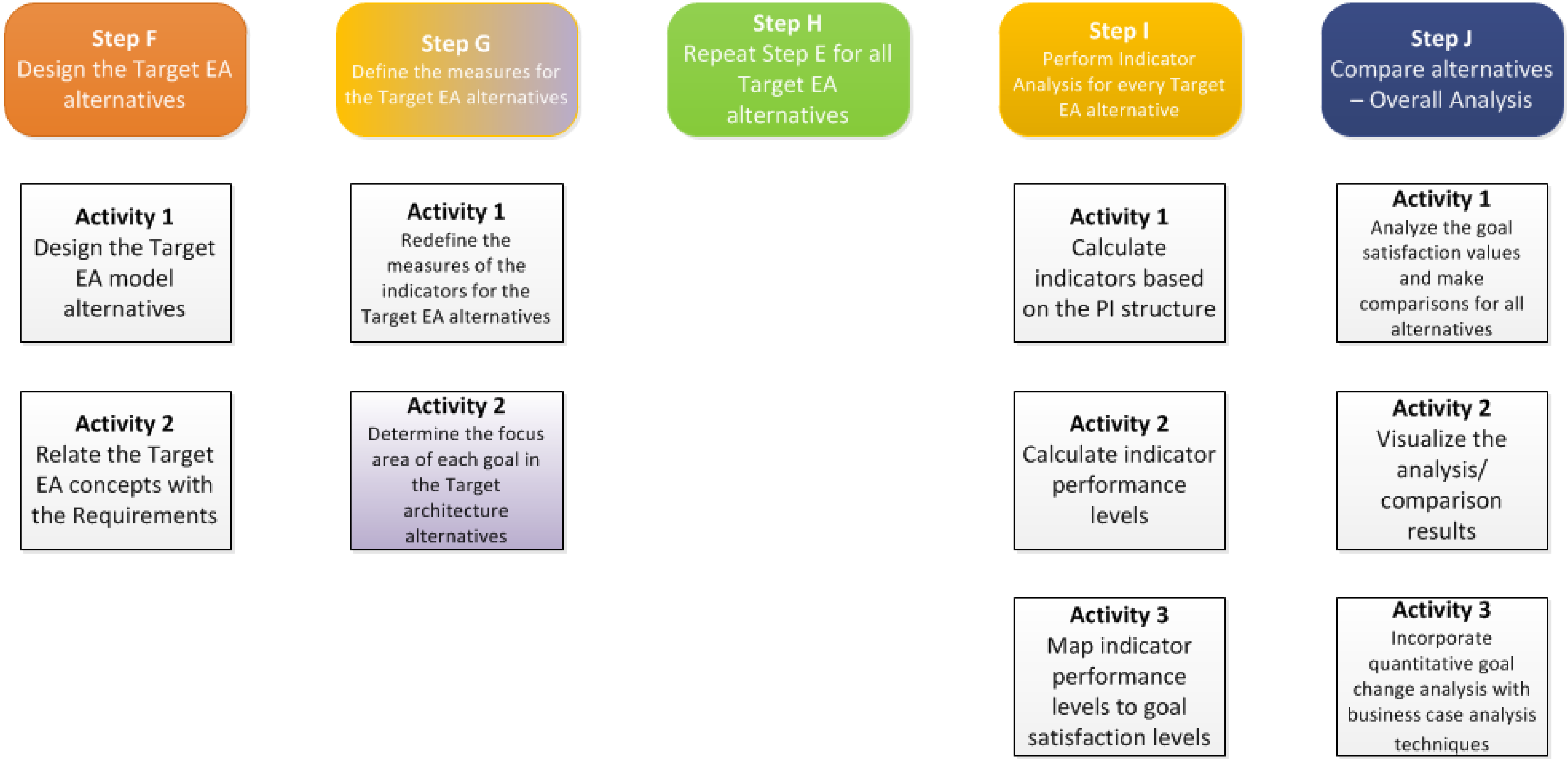


Figure 120 - EAGQ Method Steps F-J and their activities

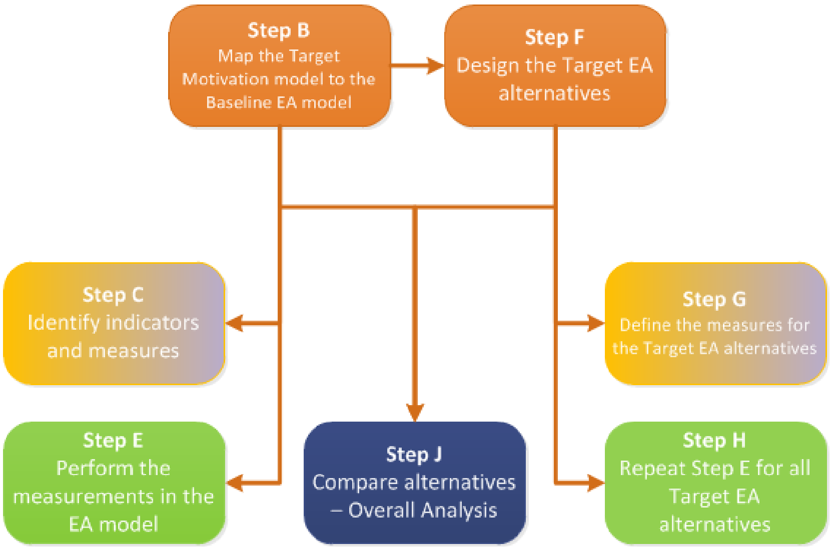


Figure 121 - Steps Dependent on EA Models and co-products

Figure 121 contains the steps that are dependent on EA models. Steps C and G have a sequential relation with Steps B and F, where the EA models are designed. They use the EA models and the relations between EA models and requirements to specify the measures derived from the models.

Except from these direct steps, Steps E and H, that are responsible for performing the measurements, receive as input the EA models for building the EA analysis models.

Moreover, Step F receives as input the requirements categorization, defined in Step B, for determining the target EA alternative models.

Finally, Step J, where the overall analysis takes place, can utilize the EA models for performing comparisons among them.

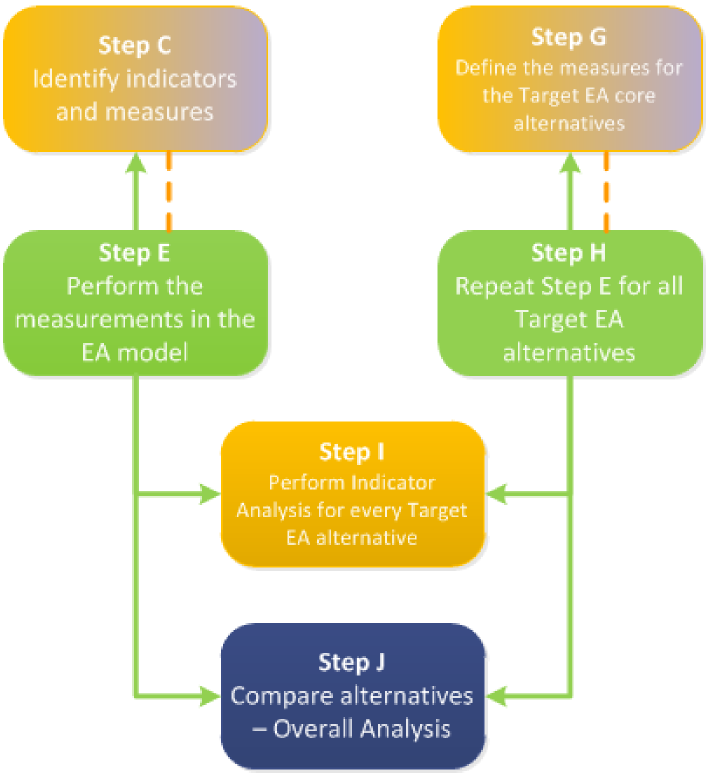


Figure 122 - Steps Dependent on Measurements and EA analysis models

Figure 122 presents the steps that are dependent on Steps H and E, where the design of the EA analysis models and the measurements take place.

Steps C and G precede the measurement-related steps, but as it is explained in the description of the steps in Chapter 5, the EA analysis models are useful for defining the measurements more accurately. This is the reason for the dashed lines, which indicate the parallel relation between activities of these steps.

Additionally, Step I receives as input the results of the measurements for calculating the indicators of the PI structure.

Finally, Step J can make use of the measurements and the analysis models for performing more detailed analysis.

Figure 123 - Steps Dependent on the Target Goal Model and the Goal Specification table

Figure 123 presents the steps that take into account the Target Goal model and/or the Goal Specification table.

Steps B and F need the Target Goal model for defining the concepts that realize the requirements and associating with them.

Step C receives as input the Target Goal model and the Goal Specification table. Both products of Step A support the indicator assignment to goals as well as the determination of the focus area of the goals. Having in mind the Target Goal model is also helpful in Step D for designing the PI structure.

In Step I, where the indicator analysis is realized, the Goal Specification table assists in the calculation of the indicator performance levels, while the Target Goal model is used for the backwards transformation of the indicator performance levels to goal satisfaction values.

Finally, Step J can include the output of Step A in its analysis.

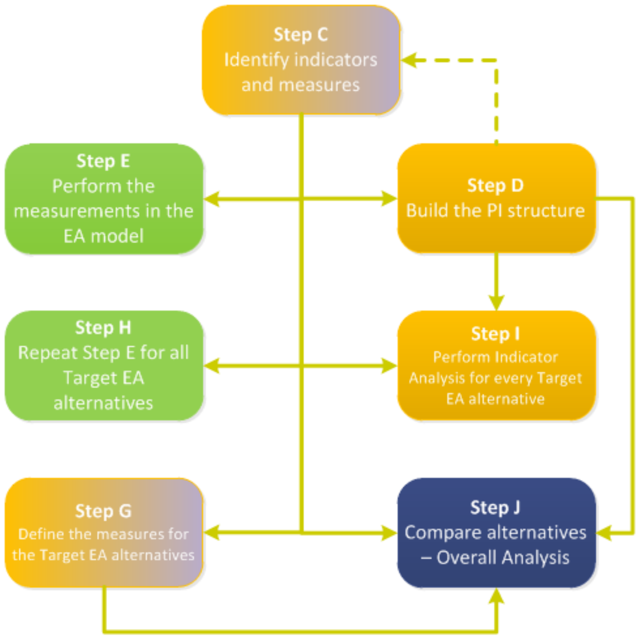


Figure 124 - Steps Dependent on the Indicator and Measure specifications and the PI Structure

Figure 124 illustrates the dependencies on the Indicator and Measure specifications and the PI Structure.

Step C, where the indicators and measures are specified is the main influencer. The specifications support the measurements in Steps E and H and the calculation and interpretation of indicators in Step I. Additionally, the specifications are used for the alignment of the measures with the target EA models in Step G.

Step D receives as input the specifications as well as the goal-indicator pairs defined in Step C for building the PI structure. The structure, in turn, is utilized in Step I for the propagation of the indicator values and in Step J for the analysis, since both absolute values and performance levels of indicators can be useful for comparing the alternatives. In Step D, the optimization of the set of measures also takes place, which leads back to Step C in case any changes need to be made (dashed line).

Finally, the overall analysis in Step G is also supported by the Indicator and Measure specifications of the baseline and target architectures provided by Steps C and G.

APPENDIX VI – EAGQ Method Specification Table Templates

Goal Specification Table template

Goal Specification Table					
	Attributes				
	Level	**Measurability**	**Evaluation type**	**Target value**	**Timeframe or**
Goal	Tree level (0 – n)	(S)oft or (H)ard	(A)chieve (C)ease (M)aintain A(v)oid (I)ncrease (D)ecrease		

Table 76 - Goal Specification Table Template

Measure Specification Table template

Measure Specification	
Name or ID of the measure	
Type of measure	
Component measures	
Type of scope	
Type of monitoring objects and points	

Table 77 - Measure Specification Table Template

Indicator Specification Table template

<table>
<tr><th colspan="2">Indicator Specification</th></tr>
<tr><td>Benefit</td><td>Benefit Specification matrix:
<table>
<tr><th>Classification change</th><th>Measurement of effect</th><th>Probability</th></tr>
<tr><td>• Do new things
• Do things better
• Stop doing things</td><td>• Financial
• Quantifiable
• Measurable</td><td></td></tr>
</table>
*time span = timeframe
*frequency of benefit = frequency of measurement</td></tr>
<tr><td>Scale</td><td></td></tr>
<tr><td>Measures</td><td></td></tr>
<tr><td>Calculation formula</td><td></td></tr>
<tr><td>Target value → pl scale = 1</td><td></td></tr>
<tr><td>Planned values</td><td></td></tr>
<tr><td>Threshold value → pl scale = 0</td><td></td></tr>
<tr><td>Tolerance value</td><td></td></tr>
<tr><td>Worst value → pl scale = -1</td><td></td></tr>
<tr><td>Performance level – Interpretation</td><td><u>Example</u>
[-1, 0] : not worthy
[0, 0.5] : progressing
[0.5, 0.8] : very good
[0.8, 1] : best</td></tr>
<tr><td>Timeframe</td><td>$$time\_difference\ (td) = \frac{t_i - t_g}{t_g}$$
Where
t_i is the timeframe of the indicator
t_g is the timeframe of the goal
Then, the final performance level is:
$$final\ pl = pl - pl * td$$
so,
if $td > 0 \Rightarrow final\ pl < pl$
if $td < 0 \Rightarrow final\ pl > pl$</td></tr>
<tr><td>Frequency of measurement</td><td></td></tr>
<tr><td>EA layer</td><td></td></tr>
<tr><td>Visualization</td><td></td></tr>
</table>

Table 78 - Indicator Specification Table Template

APPENDIX VII – Visualization types and options

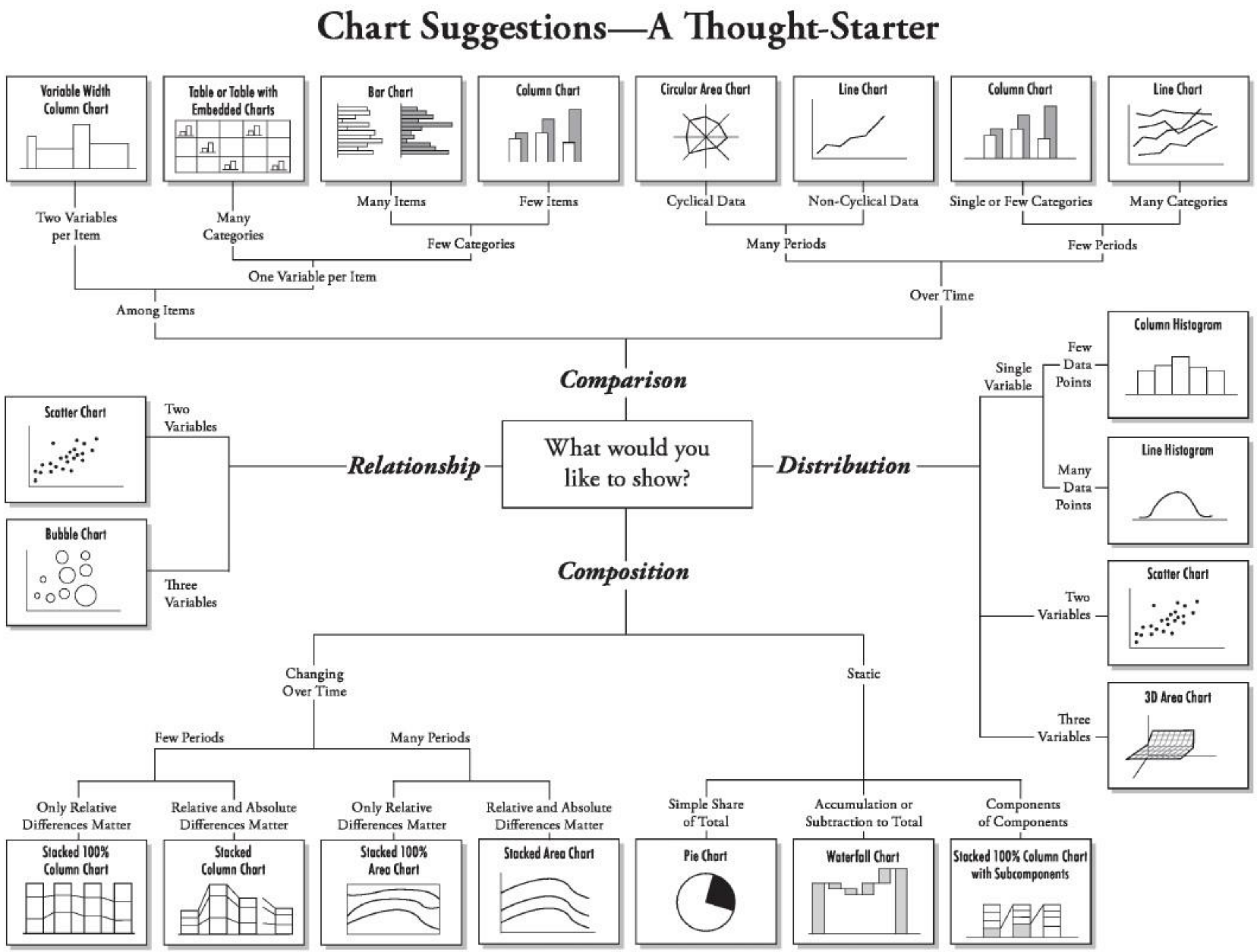


Figure 125 - Visualization types and options, obtained from Priyanto [115]

APPENDIX VIII – Demonstration of the ArchiSurance case study: Additional material

Requirements Realization in Baseline EA

Key: Requirement colors
red → *not covered*
orange → *partly covered*
purple → *covered*

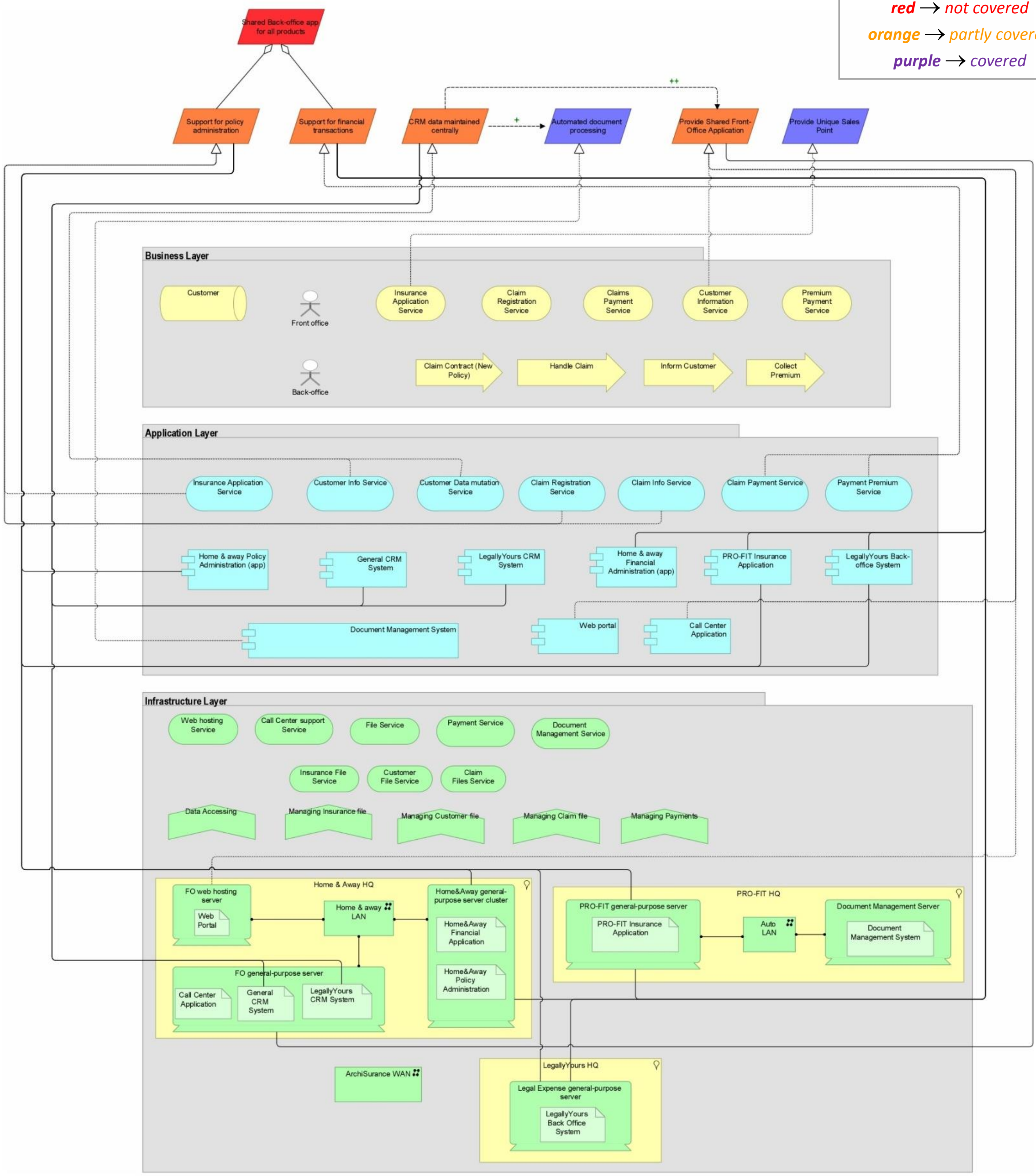


Figure 126 - Requirements Realization in Baseline EA

Indicator Calculation Formulas and Measurement approaches

Table 79 presents the calculation formula for every indicator as well as the performance levels and their interpretation. The shaded cells comprise the indicators calculated from measures. The measures and their corresponding calculation formulas as well as component measures are presented in Table 80.

Indicator	Calculation Formula	Analysis-specific attributes	Performance level –Interpretation
Profitability Index	$$PrI = \frac{IR_{total} + CR_{total}}{Investment\ Costs}$$ Where IR_{total} : Achieved revenue increase CR_{total} : Total Cost Reduction Investment Costs: Costs for implementing the Target EA	• Target value: 3% • Threshold value: 1.5% • Worst value: 1%	**Same for all* [-1, 0] : not worthy [0, 0.2] : basic/minimum [0.2, 0.5] : progressing [0.5, 0.8] : very good [0.8, 1] : best
Total Cost Reduction	$$CR_{total} = CR_{maintenance} + CR_{personnel}$$ Where $CR_{maintenance}$: Maintenance cost reduction $CR_{personnel}$: Personnel cost reduction	• Target value: 212K • Threshold value: 180K • Worst value: 130K	
Maintenance cost reduction	$$CR_{maintenance} = CR_{app} + CR_{CRM}$$ Where CR_{app} : App maintenance cost reduction CR_{CRM} : CRM maintenance cost reduction	• Target value: 190K • Threshold value: 160K • Worst value: 140K	
App maintenance cost reduction	$$CR_{app} = B_{app} - T_{app}$$ Where T_{app} : target EA application maintenance costs B_{app} : baseline EA application maintenance costs	• Target value: 120K • Threshold value: 85K • Worst value: 60K	
CRM maintenance cost reduction	$$CR_{CRM} = B_{CRM} - T_{CRM}$$ Where T_{CRM} : target EA CRM system maintenance costs B_{CRM} : baseline EA CRM system maintenance costs	• Target value: 70K • Threshold value: 45K • Worst value: 30K	

Personnel cost reduction	$CR_{personnel} = \frac{Rework_{time} + Support_{time}}{T} * W$ Where $Rework_{time}$: Rework-time reduction $Support_{time}$: Customer-support search time reduction	• Target value: 22K • Threshold value: 16K • Worst value: 12K	
Rework-time reduction	$Rework_{time} = B_{ReworkTime} - T_{ReworkTime}$ Where $T_{ReworkTime}$: target EA rework time $B_{ReworkTime}$: baseline EA rework time	• Target value: 20% * $B_{ReworkTime}$ = 6h/week • Threshold value: 17% * $B_{ReworkTime}$ = 5.1h/week • Worst value: 12% * $B_{ReworkTime}$ = 3.6h/week	
Customer-support search time reduction	$Support_{time} = B_{SearchTime} - T_{SearchTime}$ Where $T_{SearchTime}$: target EA search time per customer request $B_{SearchTime}$: baseline EA search time per customer request	• Target value: 80% * $B_{SearchTime}$ = 16.57h/week • Threshold value: 50% * $B_{SearchTime}$ = 10.35h/week • Worst value: 20% * $B_{SearchTime}$ = 4.14h/week	
Achieved revenue increase	$IR_{total} = B_{revenue} * ICS_{premium}$ Where $ICS_{premium}$: the percentage of the cross-selling premium increase $B_{revenue}$: the baseline revenue based on which the increase is to be achieved *<u>Assumption</u>: the analogy between Cross-Selling revenue and total revenue is static:* $\frac{Cross\ Selling}{Revenue} = s$	• Target value: 340K • Threshold value: 310K • Worst value: 250K	
Cross Selling premium increase	$ICS_{premium} = \frac{T_{CrossSellingpremium} - B_{CrossSellingp}}{B_{CrossSellingpremium}}$	• Target value: 30%	

	Where $T_{CrossSellingpremium}$: target EA total annual cross-selling premium $B_{CrossSellingpremium}$: baseline EA total annual cross-selling premium	• Threshold value: 22% • Worst value: 18%	
Revenue retention rate increase	$RRR_{increase} = T_{RRR} - B_{RRR}$ Where T_{RRR} : target EA revenue retention rate B_{RRR}: baseline EA revenue retention rate	• Target value: 20% • Threshold value: 15% • Worst value: 12%	

Table 79 - ArchiSurance Indicator calculation Formulas and Performance levels

Measure	Measurement method or function [Type of measure]	Component measures
Total sum of app maintenance costs	• **Baseline:** $$\boldsymbol{B_{app}} = \boldsymbol{BMaintCosts_{poladm}} + \boldsymbol{BMaintCosts_{financial}}$$ • **Target:** $$\boldsymbol{T_{app}} = \frac{\boldsymbol{TMaintCosts_{poladm}} + \boldsymbol{TMaintCosts_{financial}}}{(\mathbf{1}+\boldsymbol{r})^{\boldsymbol{t}}}$$ The *time value of money* should be taken into account for future costs, if it is not already used for the component measures. The formula used is : $$PV = \frac{FV_t}{(1+r)^t}$$	• Sum of policy administration maintenance costs • Sum of financial applications maintenance costs
Sum of policy administration maintenance costs	• **Baseline:** $$\boldsymbol{BMaintCosts_{poladm}} = \boldsymbol{direct\ sw\ costs} + \boldsymbol{sw\&hw\ maint_{labor}} + \boldsymbol{overhead\ costs} + \boldsymbol{hw\ maint\ costs}$$ • **Target:** $$\boldsymbol{TMaintCosts_{poladm}} = \frac{\boldsymbol{MaintCosts_{poladm}}}{(\mathbf{1}+\boldsymbol{r})^{\boldsymbol{t}}}$$	• Direct policy admin app costs: upgrade & license fees • Policy admin. app Personnel costs: maintenance labour • Policy admin. app Overhead maintenance costs • Policy admin. app h/w maintenance costs
Sum of financial applications maintenance costs	• **Baseline:** $$\boldsymbol{BMaintCosts_{financial}} = \boldsymbol{direct\ sw\ costs} + \boldsymbol{sw\&hw\ maint_{labor}} + \boldsymbol{overhead\ costs} + \boldsymbol{hw\ maint\ costs}$$ • **Target:** $$\boldsymbol{TMaintCosts_{financial}} = \frac{\boldsymbol{MaintCosts_{financial}}}{(\mathbf{1}+\boldsymbol{r})^{\boldsymbol{t}}}$$	• Direct financial app costs: upgrade & license fees • Financial app Personnel costs: maintenance labour • Financial app Overhead maintenance costs • Financial app • h/w maintenance costs
Sum of CRM data maintenance costs	• **Baseline:** $$\boldsymbol{B_{CRM}} = \boldsymbol{CRM\ app\ maint_{costs}} + \boldsymbol{CRM\ hw\ maint_{costs}} = \boldsymbol{direct\ sw\ costs} + \boldsymbol{sw}\ \&\ hw\ \boldsymbol{maint_{labor}} + \boldsymbol{overhead\ costs} + \boldsymbol{hw\ maint\ costs}$$ • **Target:** $$\boldsymbol{T_{CRM}} = \frac{\boldsymbol{CRM\ app\ maint_{costs}} + \boldsymbol{CRM\ hw\ maint_{costs}}}{(\mathbf{1}+\boldsymbol{r})^{\boldsymbol{t}}}$$	• CRM app maintenance costs • CRM h/w maintenance costs

Total rework-time [h/week]	$$\boldsymbol{Rework_{time}} = \boldsymbol{adminTime_{calimCheck}} + \boldsymbol{re} - \boldsymbol{processTime_{claim}}$$	• Administration time for checks of claims • Time to re-process (response time)
Average Search time per customer request [h/week]	$$\boldsymbol{SearchTime} = \boldsymbol{\#search_{queries}} * \boldsymbol{responseTime_{query}} * \boldsymbol{f_{cc}} * \frac{\mathbf{5}}{\mathbf{3600}}$$ f_{cc} : arrival frequency of customer requests at the call center per day 5 working days 3600 ⇒ turn seconds into hours	• # of search queries per customer request • Response time per search query
Total annual premium from customers with more than one contracts	• **Baseline:** $$\boldsymbol{BCrossSelling_{premium}} = \boldsymbol{Annual\ reveune\ from\ customers_{>1}}$$ • **Target:** $$\boldsymbol{TCrossSelling_{premium}} = \frac{\boldsymbol{Annual\ reveune\ from\ customers_{>1}}}{(\mathbf{1}+\boldsymbol{r})^{\boldsymbol{t}}}$$	
Revenue retention rate	$$\mathbf{RRR} = \sum_{\mathbf{i}} \mathbf{CRR_i} * \mathbf{RetainedValue\%_{CustGroup_i}}$$ *where* **Customer Retention Rate:** $$\boldsymbol{CRR} = \frac{\#endCustomers - \#newCustomers}{\#beginningCustomers} * 100\%$$ $$\boldsymbol{CRR_i} = CRR * CustGroup\%_i$$ $\boldsymbol{CustGroup\%_i}$: percentage of customers that have retained their prior premium value by the same x% $\boldsymbol{RetainedValue\%_{CustGroup_i}}$: the retained x% of the prior value	• # new customers, during the year (period) • # beginning customers, at the beginning of the year (period) • # end customers, at the end of the year (period) • Total annual premium per customer: previous year (period) • Total annual premium per customer: current year (period)

Table 80 - ArchiSurance Measures and Calculation formulas

Applying the ABC Methodology: Policy administration application maintenance costs in Baseline EA

Using the ABC methodology would be convenient for calculating the exact costs for every application component. Assuming that the direct costs are easy to assign to each application component individually, such as upgrade costs, license fees, labor costs both for s/w and h/w that support the applications; this task is not easy for the overhead or indirect costs. The overhead costs and indirect costs can be estimated in total, but the allocation to each application and consequently to each business unit or function needs further analysis.

Overhead or indirect costs include downtime costs, documentation costs, administrative costs, etc. which can be calculated, for example, based on costs per hour by using labor costs and downtime hours, documentation hours and administrative hours (e.g. from the operating costs).

To apply the ABC method for allocating the overhead costs to the three application components and consequently the three business units, the following concepts need to be identified:

- The three application components are the **cost objects:**
    - Home & Away Policy Administration
    - Legally Yours Back-office system
    - PRO-FIT Insurance application
- The total amount of overhead maintenance costs comprises the **resources**
- The **activities or cost categories** are: the downtime costs, documentation costs, administrative costs and depreciation costs
- The **resource driver** (coefficient for assigning the overall overhead maintenance costs to each activity or cost category) is:
    - Application usage: # policies processed (temporal scope: since the merger)
- The **activity driver** is: cost per policy processed

By identifying the exact activities and defining costs per application, an evaluation of the costs generated per business unit could be achieved. This analysis is more detailed and can directly indicate the costs per policy or claim for each business. Having this information, a decision could be made regarding the cost allocation per business unit or regarding the value of each application comparing to its costs, so that, for instance, actions could be taken for equalizing the costs.

In the example, the overall costs of all three applications are important and there is no need for further decomposition and re-allocation of them, since the total costs are going to be compared with the future alternatives. Thus, the ABC methodology is not going to be analyzed and applied in this example. Its application is optional. In the demonstration, the overhead costs are considered known for each individual application and infrastructure component.

An example for applying the ABC method is provided by Rchid et al. [131] for the aircraft maintenance service costs.