Development of a Project Portfolio Selection Method that includes Enterprise Architecture Complexity as a Criteria

G.M. Cooiman
Master Thesis
April 2021

Study Programme
MSc Business Administration
Track
Digital Business

Graduation Committee
Prof. dr. M.E. Iacob
Dr. R. Effing
External Supervisor
Ir. S.W. Nijenhuis

University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Development of a Project Portfolio Selection Method that includes Enterprise Architecture Complexity as a Criteria, Master Thesis, University of Twente, April 2021

AUTHOR:
G.M. Cooiman
Study Programme: MSc Business Administration
Email: g.m.cooiman@student.utwente.nl

Prof. Dr. M.E. Iacob
Faculty: Behavioural Management and Social Sciences
University: University of Twente, Enschede, The Netherlands
Email: m.e.iacob@utwente.nl

Dr. R. Effing
Faculty: Behavioural Management and Social Sciences
University: University of Twente, Enschede, The Netherlands
Email: r.effing@utwente.nl

Ir. S.W. Nijenhuis
Company: Fortes Solutions B.V.
Position: Director
Email: s.nijenhuis@fortes.nl

MANAGEMENT SUMMARY

Situation
Organizations are faced with a rapidly changing environment and an ever-increasing dependency on their IT infrastructure. Change is initialized through initiatives in the form of for example projects. Several criteria are important to measure and weigh in the selection process. Costs, benefits, and risks are the most common criteria in the selection process.

Complication
Despite the organization's best efforts to get a realistic judgment on the projects, still, a lot of projects seem to fail. Failing can be put as exceeding the budget, overestimating the returns, and not reaching the deadlines. Literature and practice show that the complexity of an organization's architecture, their so-called "Enterprise Architecture" has a significant influence on the success of a project. However, this factor is often neglected in the project selection process.

Question
The goal of this research is to develop a project selection method that uses enterprise architecture complexity as a criterion in the selection process. The key research question of this thesis is: *How to design a Project Portfolio Selection Method that uses Enterprise Architecture Complexity as a decision criteria?*

Answer
This master thesis introduces an enterprise architecture complexity analysis approach based on a multi-criteria decision analysis methodology in the context of a project selection method. It presents a step-by-step guide to get from enterprise architecture and a list of possible projects to a prioritized list of projects based on weighted scores for Enterprise Architecture Complexity and Enterprise Architecture based costs, risks, and benefits.

Methodology
This research is approached as a design science research and used the design science research methodology as proposed by Peffers et al. to structure this master thesis.

Chapter 2 contains a thorough literature review to answer the questions regarding the state of the art of project portfolio selection and enterprise architecture. The analysis methods for the proposed method are selected in this chapter as well. Chapter 3 presents the proposed method, including detailed explanations of the steps and analysis methods. Chapter 4 presents an application of the proposed method in practice at a company specialized in project portfolio management software. Chapter 5 presents the results of the evaluation of the proposed method in which five experts were interviewed. Chapter 6 concludes the thesis with an overall discussion and it acknowledges the contributions, limitations, and future work.

Conclusion
The application in practice and evaluation shows that the method has potential and could be used in practice. The quantitative approach towards complexity reduces the subjectivity that is currently evident in the selection process of a lot of organizations and opens up the organization for a whole new discussion. The enterprise architecture complexity analysis can be used separately from the rest of the model and is therefore suitable for adoption in quantitative project selection methodologies that are used by organizations. The aspects of the model that were questioned during the evaluation are presented in chapter 6 and need to be taken into account.

TABLE OF CONTENTS

Table of Synonyms

Acronym	Description
AHP	Analytical Hierarchy Process
BWM	Best-Worst Method
DevOps	Development and Operations
EA	Enterprise Architecture
EAC	Enterprise Architecture Complexity
EACM	Enterprise Architecture Complexity Metrics
IT	Information Technology
MAUT	Multi Attribute Utility Theory
MCDA	Multi Criteria Decision Analysis
PMO	Project Management Office
PPM	Project-Portfolio Management
PPSP	Project Portfolio Selection Process
SLR	Structured Literature Review
UTAUT	Unified Theory for Acceptance and Use of Technology

Table of Tables

Table of Figures

1 INTRODUCTION

Organizations are facing a rapidly evolving environment, and have to keep innovating and changing their business to stay competitive (Elbok, 2017; Beese, 2016; Gellweiler, 2020). The Agile way of working and the integration of development and operations (DevOps) are reducing the time to market of innovations (Kaisler et. Al., 2005; Maciaszek et. Al., 2006). This requires organizations to choose their organizational change initiatives effectively, and efficiently. Projects (business change initiatives) are considered an important instrument to achieve these organizational goals (Aldea et. Al., 2019; Iamratanakul et. Al, 2008).

The role of Project Portfolio Management (PPM) is to evaluate, select, and prioritize new projects, as well as to revise priority, and possibly eliminate and reduce projects in progress (Danesh et al., 2015). Padovani and Carvalho (2016) also stated that PPM is an emerging aspect of business management that focuses on how projects are selected, prioritized, integrated, managed, and controlled in the multi-project context that exists in modern organizations. PPM not only deals with the selection of the projects, but also consists of elements to support overall portfolio management, such as portfolio optimization, portfolio approval, and portfolio evaluation (Aldea et al., 2019). The Project Portfolio Selection process is particularly relevant in this research, as it deals with choosing the projects that fit the best with organizational goals. Therefore, it will be the main focus of this research.

By definition, Project Portfolio Selection is the periodic activity involved in selecting a portfolio of projects, that meets an organization's stated objectives without exceeding available resources or violating other constraints (Ghasemzadeh & Archer, 2000). Project Portfolio Selection is a complex process, that involves various factors that need to be taken into account. The factors that are mostly discussed in Project Portfolio Management literature include the expected cost, associated risk, and the benefits of a project (Aldea et. Al., 2019; Iamratanakul et. Al, 2008). Organizations only have a limited amount of resources that can be used for the execution of projects. Therefore, the selection of a project to fulfill certain organizational goals should be made in such a way, that it is successful and supports the organizational goals. The process of aligning business with IT is called Business and IT Alignment (Zhang, 2018).

The concept of business-IT alignment is one of the most examined topics in academia and real-life (Ullah & Lai, 2013). Business-IT alignment contributes to value generation from IT investments (Henderson & Venkatraman, 1993). Enterprise Architecture is considered to be an effective methodology for business-IT alignment (Bhattacharya, 2018), which deals with the interrelationship of IT and business to attain strategic goals (Ullah & Lai, 2013) and to create business value (Mosthaf & Wagner, 2016). Both functions strategically align to the business and control subordinate functions on the tactical level to maintain consistency for future changes. While EA concentrates on IT projects, PPM encompasses all major changes of the enterprise. Both sides analyze potential projects based on their needs. (Gellweiler, 2020)

Research has shown that IT projects are often not successful (Antlova, 2010). Complexity is recognized by Lucke et al. (2010) and Daniels and Lamarsh (2007) as a significant factor for the lack of success of IT projects. The Dutch Tax Office is an example that involved a very complex application landscape with unsuccessful effects of their one billion dollar IT change program[1]. Budget overruns,

[1] https://www.computerweekly.com/news/252444408/Dutch-government-IT-projects-run-1bn-over-budget

missed deadlines, unfulfilled requirements, and overestimated returns are results of many failed projects. However, in the current Project Portfolio Selection Models complexity is not taken into account at all (Iamratanakul et. Al, 2008), although it is recognized as a major factor for project success.

Previous research also proved that Project Portfolio Selection Models are not complete in their current form. Aldea et al. (2019) showed that Capability Based Planning is a complementary element for Project Portfolio Selection which addresses the functional aspects of Enterprise Architecture by modeling it using widely used Enterprise Architecture modeling techniques. This research aims to create a Project Portfolio Selection method with Enterprise Architecture Complexity as a criteria. However, this research does not look at the functional aspects of the Enterprise Architecture, it focuses on the quality aspects of the Enterprise Architecture. Organizations should assess their Enterprise Architecture during the Project Portfolio Selection process, analyze the impact of the projects on the Enterprise Architecture, and use this information in addition to other selection criteria. This could help an organization choose the best project based on its strategic needs.

To develop a Project Portfolio Selection method that improves the success of IT projects, the analysis of the complexity of the Enterprise Architecture and the projects, along with analytical techniques are integrated. The main objective of this research is therefore the design of a method for project selection that integrates a Project Portfolio Selection Method with Enterprise Architecture Complexity. The method should help organizations determine which projects to do based on multiple important criteria.

1.1 Research Goal

A major factor in the success of IT projects is recognized as the complexity of the Enterprise Architecture (Khosroshahi et al., 2016) However, the complexity of the Enterprise Architecture is not recognized as a factor in existing Project Portfolio Selection methods (Gellweiler, 2020; Costa, 2020). The purpose of this research is the development of a method that supports current project selection methods by integrating them with Enterprise Architecture Complexity. By using that input in the project selection criteria, portfolio managers can select projects based on the added complexity to the architecture. As the consequences of certain projects are currently not tangible for portfolio managers to base their decision on, this addition will provide more insight into the effects of the project on the EA. In this way, the added complexity is anticipated, such that undesired increases in complexity can be prevented, resulting in lower failure rates in IT projects.

1.2 Research Scope

Currently, Project Portfolio Selection methods are mostly limited to the known criteria such as expected costs, benefits, and risks. Costs and risks should be minimized, while benefits should be maximized. 80% of IT projects are not successful. Therefore we can imply that current practices are not complete enough to indicate whether the project will be successful. The complexity of Enterprise Architecture is recognized as an important factor for project success. The purpose of this research is the development of a method that supports current project selection methods by extending common selection criteria with a way of defining added EA complexity by a project. Complexity is a broad concept that is non-consensual in certain aspects. To address its broadness, and to scope this research to only the relevant aspects for PPM with regards to complexity, a survey will be conducted. This survey will be conducted among experts in the field of PPM. Fortes Solutions is a company that supports international clients, from NGOs to large enterprises, with their software and expertise in portfolio management. The survey will be conducted among its consultants, partners, and key employees that have practical and theoretical knowledge of PPM. The consultants are all stationed in the Netherlands but work for clients all over the world. The partners are also stationed abroad. Their expertise is used to select areas of complexity that are most relevant for PPM. Those areas are selected and used in the rest of the thesis. This research started officially in May 2020. The proposal phase restarted in October 2020 and finished in February 2021. The execution of the research is planned from February 2021 until the end of April 2021. The research is executed in the Netherlands, which is also the location of the headquarters of Fortes Solutions, and the location of the research institute of the writer.

1.3 Research Design

1.3.1 Research Questions

The main research question is:

How to design a Project Portfolio Selection Method that uses Enterprise Architecture Complexity as a decision criteria?

The following sub-research questions are derived from the research question. First, it is necessary to define the state of the art of both Project Portfolio Management and Enterprise Architecture. As this research is focused on the Project Portfolio Selection process, we focus our research there. As well as for Enterprise Architecture, where the focus will be on defining what complexity means in the context of Enterprise Architecture. Literature also needs to show what existing Project Portfolio Selection method is most suitable for our proposed extension with Enterprise Architecture Complexity. Therefore, the following sub-research question is derived:

1. What is the current state of the art of Project Portfolio Management and Enterprise Architecture, and what Project Portfolio Selection Method is most suitable for integration with Enterprise Architecture Complexity?

Secondly, based on the results of the previous research question, a Project Portfolio Selection method should be designed with Enterprise Architecture Complexity as a criteria. This is a design problem. The following sub-research question is derived:

2. How to design a Project Portfolio Selection Method with Enterprise Architecture Complexity Metrics?

Lastly, the developed method should be tested, evaluated, and validated. Therefore, the following sub-research question is identified:

3. How can the proposed method be validated?

1.4 RESEARCH PROCESS

The research questions defined in Section 1.3.1 and 1.3.2 include knowledge questions, as well as a design problem, namely the design of a method. The descriptive research is approached with a Systematic Literature Review by using the guidelines that are proposed by Kitchenham and Charters (2007). The design science research part is approached using the Design Science Research Methodology of Peffers et al. (2006). This method is shown in Fig 1.

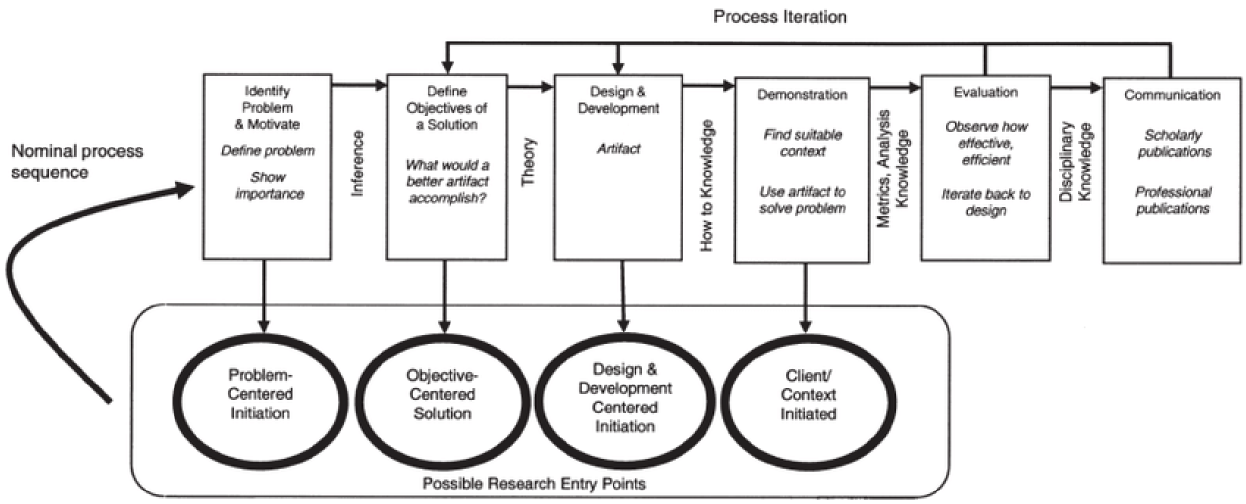


Fig. 1. DSRM Process Model (Peffers et al., 2007)

The research process is mapped on the DSRM Process model (Peffers et al., 2007) as described below.

Model Step	Description
Problem identification and motivation	The initial step includes the identification of the problem and proposes the solution for the problem. The motivation for the research is explained and the research questions are derived. This part is included in Chapter 1.
Define the objectives for a solution	With the defined problem and motivation for the research, the objectives of the research can be defined next. The objectives in this part should provide a logical set of steps to build upon. It operates as a kind of roadmap that provides a template structure for the research outputs. This part is included in Chapter 1 and Chapter 2.
Design and development	The design and development of the method are defined in Chapter 3.
Demonstration	The demonstration of this research is structured as a single case study. The case study will be performed at a software company called Fortes. Fortes is a midsize independent software vendor that is currently in a digital transformation. This part is included in Chapter 4.
Evaluation	The evaluation of this research consists of interviews with experienced employees at Fortes using the format of the Unified Theory of Acceptance and Use of Technology (UTAUT). This part is included in Chapter 5.
Communication	The last step of DSRM involves communication. The communication part in this research is performed through the thesis defense when the thesis is finalized and submitted.

Table 1, Research Structure

1.5 Research Structure

This report is structured as defined by Peffers et. Al. (2006). It follows the steps that are defined in DSRM. In general, the thesis can be divided into six parts, namely Introduction, Literature Review, Design and Development, Demonstration, Evaluation, and Conclusion.

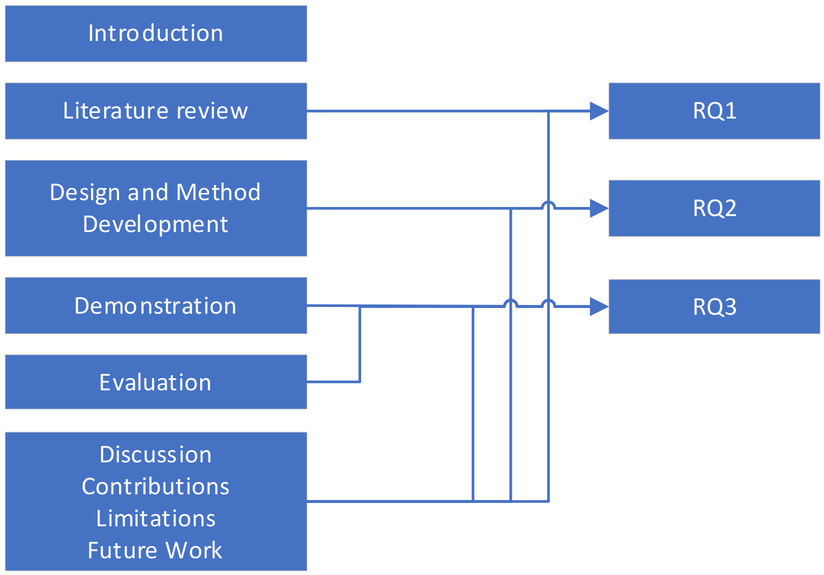


Fig 2. Research Structure

2 LITERATURE REVIEW

In this section, the knowledge questions that are identified in section 1.3 are answered. The research method is explained, followed by the design of the literature review. Thereafter the review conduction will be explained through the inclusion and exclusion criteria, the quality assessment, backward search, and synthesis. The results of this literature review are handled, concluded, and discussed afterward.

2.1 RESEARCH METHODOLOGY

Systematic Literature Review (SLR) has been chosen as the research method. This study will use the guidelines that are proposed by Kitchenham and Charters (2007). According to Kitchenham and Charters (2007), a systematic literature review process consists of three consecutive stages: planning, execution, and result analysis; and another stage which is performed throughout the whole process to store the results of the previous stages: packaging. Therefore, there are two checkpoints in the course of the process to evaluate that the systematic literature review process executed is correct. Fig. 1 outlines all the activities included in each phase that will be described in detail in the following subsections.

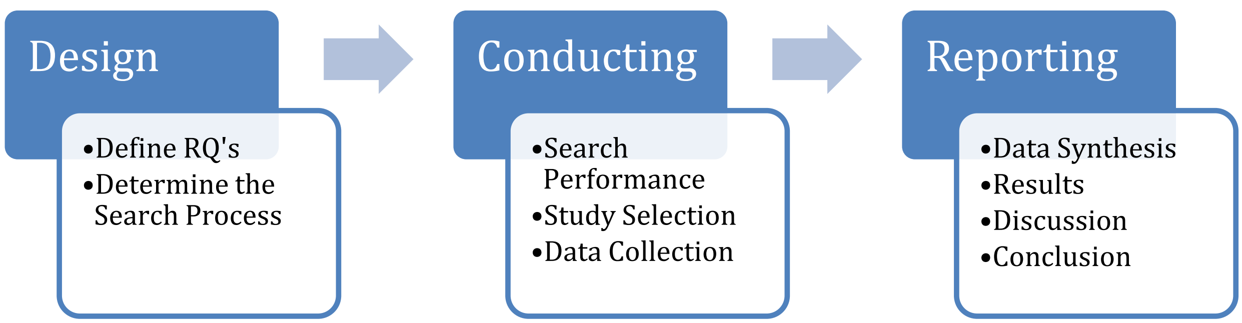


Figure 3. SLR design (Kitchenham and Charters, 2007)

2.2 REVIEW DESIGN

This section describes the foundation of the SLR by defining the research questions and their accommodating keywords. Also, the search process will be explained. The review is conducted between December 2020 and January 2021.

2.2.1 SLR research questions

The research questions are defined in section 1.3. The first sub-question is a knowledge question, and therefore will be evaluated in this structured literature review. The question is: *"What is the current state of the art of Project Portfolio Management and Enterprise Architecture, and what Project Portfolio Selection Method is most suitable for integration with Enterprise Architecture Complexity?".* The scope of this question is quite broad, and therefore this question will be split into multiple sub-questions.

1. What is the state of the art of Project Portfolio Management, with Project Portfolio Selection Methods in particular?
2. What is the state of the art of Enterprise Architecture Complexity?
3. What Project Portfolio Selection Method is most suitable for an extension with Enterprise Architecture Complexity?
4. What metrics are used to measure complexity in Enterprise Architecture?

The relevant keywords for this structured literature review can be found in Table 2.

Keywords
Enterprise Architecture, EA, Project Portfolio Management, PPM, Portfolio management, Project management, Alignment, complexity, methods, tools, techniques, quantification, selection, project selection, selection

Table 2, Research Question Keywords

2.2.2 Search process

Scientific databases are used to find peer-reviewed literature from proper journals with relevance to this research. The following databases are used in this research:

- Scopus (https://www.scopus.nl)
- Web of Science (https://www.webofknowledge.com)
- IEEE Xplore (https://www.ieeexplore.ieee.org)

Scopus is the largest database of these two but lacks some research in the social sciences. Therefore Web of Science is added to have good coverage on both technological and business/social research. The following search query is executed on the databases:

((“Enterprise Architecture” OR ea OR Enterprise Architecture management) AND(“project portfolio management” OR “portfolio management” OR “project management”)) AND (selection OR complex* OR method* OR tool* OR technique* OR planning OR metrics OR quanti*)

The query specifically looks at research that covers both the areas of project portfolio management and Enterprise Architecture. This approach is chosen by the researcher to narrow down the results to articles that cover both Project Portfolio Management and Enterprise Architecture. The query was tested on the selected databases such that these subjects would also come up separately, but that resulted in a lot of noise. The articles that combined the subjects were more relevant and recent.

2.3 Review Conduction

This section discusses the steps for the conduction of this research. This includes the inclusion and exclusion criteria (section 2.3.1), forward and backward search (section 2.3.2), and synthesis (2.3.3).

2.3.1 Inclusion and exclusion criteria

A lot of results were found irrelevant for this research. Therefore, some criteria are defined to filter out the irrelevant articles. The studies should be reported in English, as other languages are not readable for the researcher, and translations could lead to misinterpretations of its contents. The studies should be published in journals, a chapter of a book, or part of conference proceedings, as these are credible sources of information. The content of the research should be relevant to at least one of the sub-research questions that are defined in section 2.2.1.

Papers that are excluded are the ones that do not meet the inclusion criteria. For example, studies that are not related to one of the research questions, or are typed as conference review, note, or short paper. Inaccessible studies are also excluded. Some papers were only accessible with a certain subscription, or in exchange for money. The three used databases contained some duplicate studies. These were merged after applying the above-mentioned criteria.

The inclusion and exclusion criteria do not contain any constraints on the year of publication. The researched databases all contain articles from after 1990. As the discussion of alignment started around that year, no constraints are given.

2.3.2 Selection Process

The selection process follows the following steps. First, the specified databases are queried with the selected keywords. Irrelevant articles are excluded based on the defined inclusion and exclusion criteria. The remaining articles are merged and deduplicated. The next step in the process is excluding articles based on the title of the article and its abstract. The list that remained was filtered on accessibility; inaccessible studies are removed. The full-text versions of the articles are evaluated. The references of the articles are also evaluated on relevance. The articles that could be in the scope of the research, but were out of the scope of the search query on the selected scientific databases were included.

2.3.3 Synthesis

Figure 3 shows the number of sources found based on the keyword search in the selected database. The second column shows the result after applying the exclusion and inclusion criteria. The number shows the number of papers that remained after applying these criteria. The third row shows the number of papers that remained after removing duplicate studies from the results. Some studies were found in more than one database and were therefore merged. Each following column shows the number of papers that remained after each defined subprocess.

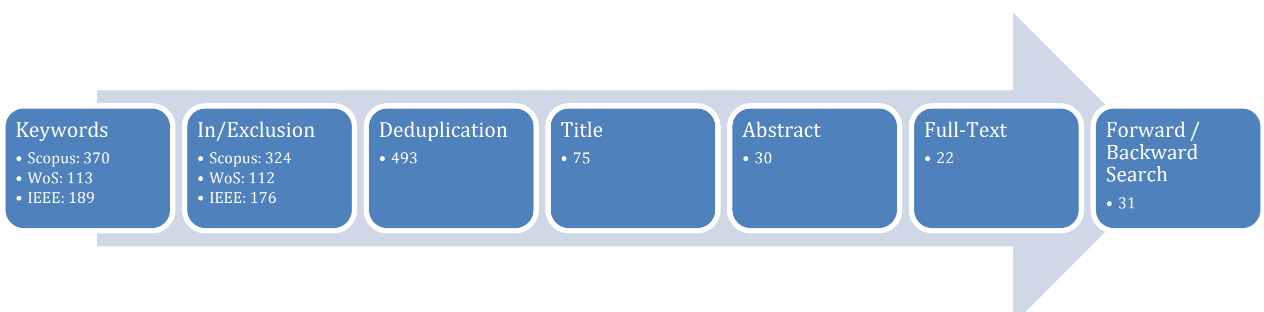


Figure 4. Literature Research Synthesis

2.4 Review Results

This section represents the findings and discussion of this review to answer the defined SLR research questions. This section is structured such that each subsection answers a specific research question, as defined in section 2.2.1.

2.4.1 Findings on RQ1

This section describes the findings of this literature review on the first research question: "*What is the state of the art of Project Portfolio Selection Methods?".*

To answer this question, we first analyze the literature and define what the definitions are, starting with Project Portfolio Management. Project Portfolio Selection is recognized as a subprocess of Project Portfolio Management, as represented by the Project Management Institute in Figure 4, and will therefore be introduced and analyzed in the subsequent section. That section presents multiple Project Portfolio Selection methods.

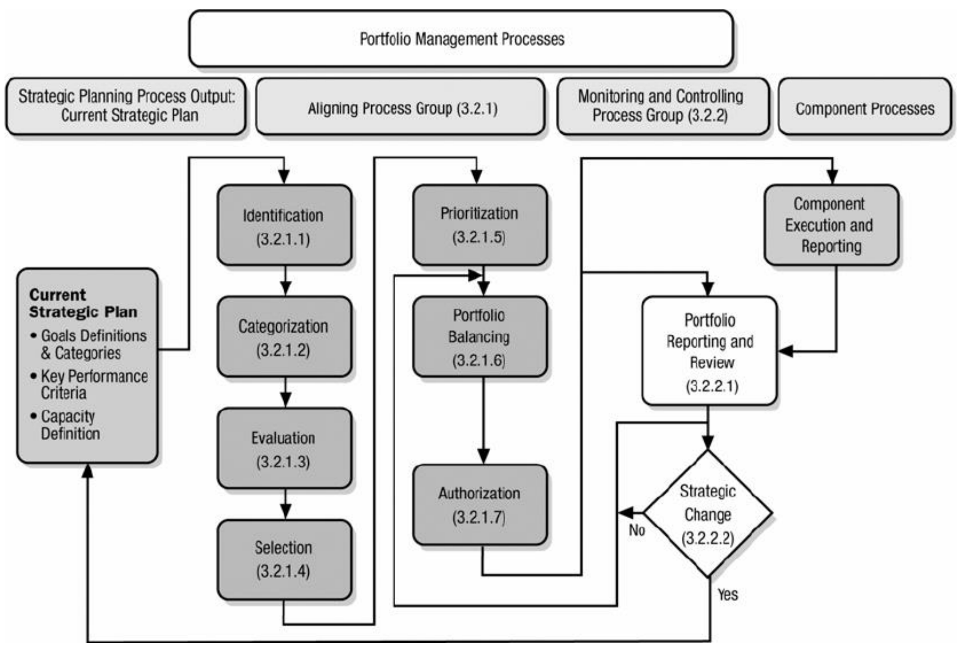


Figure 5. Portfolio Management Processes (PMI, 2013)

2.4.1.1 Project Portfolio Management

For business competitiveness, organizations must master the definition and implementation of their strategies. However, the best strategies can be useless without proper implementation (Unit, 2013). Project Portfolio Management (PPM) is embedded in the organization's overall strategy to accomplish objectives and realize the strategies of an enterprise (PMI, 2013, pp. 5–7). PPM strives for optimal resource and budget allocations and preschedules projects to best accomplish the organization's goals (Cooper, Edgett, & Kleinschmidt, 1999, p. 334).

Projects can be thought of as business change initiatives that compete for resources and monetary funds; these demands must be monitored and decided if deviations from the cost baselines occur as projects progress (Gellweiler, 2020). Projects must add value to organizations and realize the expected return on investment. Although the Project Management Office (PMO) supports the alignment of projects with the organization's strategy (Otra-Aho et al., 2019), projects tend to have a weak alignment with the business strategy because most of them are conceived to solve urgencies in operations or to answer to senior managers' specific requirements (Anyosa Soca, 2009), (Garcia et al. 2018).

The terms that are introduced have many available definitions and interpretations. The most frequently used definitions are however originating from the Project Management Institute (2013). These definitions will be used in this research and can be found in Section 9.3.

2.4.1.2 Project Portfolio Selection Methods

Project portfolio selection is understood as a dynamic decision-making process to evaluate, select and prioritize a project or a set of projects for implementation through the allocation of constrained resources and alignment with corporate strategies (Archer & Ghasemzadeh, 1999; Cooper, R. et al., 2001). Therefore, project portfolio management includes the Project Portfolio Selection process as explained by Archer et al (1999) and Cooper et al (2001).

An overview of available Project Portfolio Selection methods was presented by Iamratanakul et al (2008). They formulated a comprehensive description of all the Project Portfolio Selection models that were available until 2008. They categorized their findings into six categories, namely Benefit Measurement Methods, Mathematical Programming Approaches, Cognitive Emulation Approaches, Simulation, and Heuristics Models, Real Options, and Ad Hoc Models. This overview is based on the findings of Iamratanakul et al (2008),

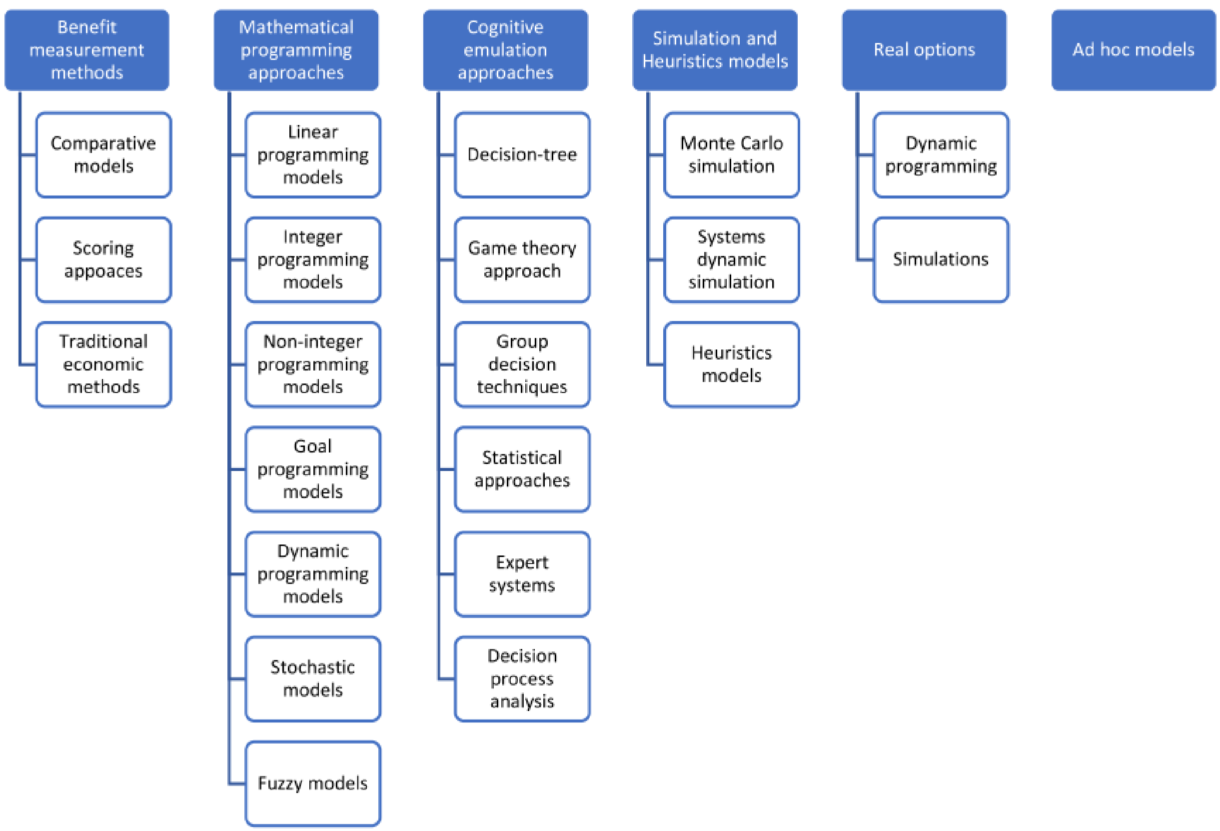


Fig. 6, Models for Project Portfolio Selection (Iamratanakul et al., 2008)

For the analysis of different project selection methods, this research will use the classification of project selection models that are defined by Ishizaka and Nemery (2013), who distinguish four types of Project Portfolio Selection models. These models are Multi-Criteria Decision Analysis (MCDA), Constrained Optimization, Scoring Models and Linear Programming. MCDA methods evaluate multiple criteria in decision making. The scoring models are methods for ranking candidate projects relative to one another. The linear programming models are quantitative tools for project portfolio selection using linear programming (LP).

Decision problems are faced by people daily. Most of these times, taking only one criterion into account is not enough. To make an informed decision, multiple criteria need to be taken into account. MCDA is developed to solve the issue of facing a lot of criteria. MCDA is an approach. Multiple methods have been developed to provide priorities or rankings on the given alternatives (Ishizaka & Nemery, 2013). Research has shown that MCDA is a useful tool to support decision-making. MCDA methods are suitable for a decision problem such as a Project Portfolio Selection. Therefore, this research will analyze what MCDA method is best suited for the method that is developed in Section 3. The analysis of what MCDA method is best suited for the method that is developed in this research will be continued in Section 2.4.3.

2.4.2 Findings on RQ2

This section describes the findings of this literature review on the second research question: *"What is the state of the art of Enterprise Architecture Complexity?".*

To answer this question, we first analyze the literature and define what the definitions are followed with an introduction on the topic through reviewing relevant and recent papers, starting with Enterprise Architecture. Enterprise Architecture can be modeled through several methodologies. The method that is used most in practice will be introduced in the subsequent section. The research question is focused on the complexity of Enterprise Architecture. Therefore, we first have to elaborate more on what Enterprise Architecture is before we can do a deep dive into what complexity is regarding Enterprise Architecture. Complexity, in general, is introduced in subsection 2.4.2.3. The dimensions of complexity in regards to Enterprise Architecture are introduced in the subsequent section.

2.4.2.1 Enterprise Architecture

The concept of architecture is quite general. It is often dependent on the context and the discipline of what architecture means. The International organization for Standardization (ISO) creates definitions and standards that are widely applicable, so for the case of Architecture, the definition of ISO is adopted in this research.

Definition: Architecture: The fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution (ISO/IEC/IEEE, 2011).

The definition is widely applicable but dependent on the system and environment. In this research, the environment is organizations. This could also be a superset of organizations or a specific department of an organization. The environment can thus be phrased as an enterprise.

Definition: Enterprise: Any collection of organizations that has a common set of goals (The Open Group, 2011).

Let's elaborate on the environment that is relevant in this research; the Enterprise Architecture.

Definition: Enterprise Architecture: A coherent whole of principles, methods, and models used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure (Lankhorst et al. 2017).

Over the last years, the field of Enterprise Architecture has seen considerable developments. The toolbox of the enterprise architect nowadays comprises a wide array of methods, techniques, and tools, such as TOGAF, the Zachman framework, and the Department of Defense Architecture Framework (Malta & Sousa, 2010). While there are many similarities in these frameworks, TOGAF is indicated by several studies as the most used by practitioners (Alwadain, Fielt, Korthaus & Rosemann, 2014)(Obitz & Babu, 2009). This is partly due to TOGAF being the only framework that is supported by a formal language, namely ArchiMate. ArchiMate is used as modeling language in this research. A description can be found in Chapter 8. A study by Slot, Dedene, and Maes (2009), based on an analysis of 49 projects, clearly shows the benefits of enterprise and project architecture. So having a good Enterprise Architecture practice may deliver direct and indirect cost savings and other benefits, because decisions are made in context: it offers a holistic view, showing the interdependencies between different parts of the enterprise. Architecture forms a strategic instrument in guiding an organization through a planned course of development. EA creates links between business architectures and IT architectures and verifies their integrity (Helfert, Doucek, & Maryska, 2013, p. 73). It also identifies business processes, applications, data, and technology (Strano & Rehmani, 2007, p. 392). EA is, however, prone to change due to e.g. new technologies, business developments, strategy changes, compliance, or new demands (Langermeier, 2018) The current EA is not capable to keep track of the pace of change in an organization and keep the architecture up-to-date.

2.4.2.2 EA Complexity

According to Davis and LeBlanc (1988), the complexity of application architecture is the "number of its components or elements, kind or type of elements and structure of the relationship between elements". Henningsson and Hanseth (2011) that is "the dramatic increase in the number and heterogeneity of included components, relations, and their dynamic and unexpected interactions in IT solutions". Also, Schneberger and McLean (2003) think similarly: "The complexity can be defined based on the number and variety of components and interactions plus the rate of change of these". These definitions would suggest that the complexity of an Enterprise Architecture would be based on the number of components, their connections and interactions, and their variety.

Schneider, Zec, and Matthes (2014) developed a framework that comprehends EA complexity in four dimensions. They identified that complexity is composed of four opposing notions of complexity. These dimensions are Organized vs Disorganized; Qualitative vs Quantitative; Subjective vs Objective; and Structural vs Dynamic. Schneider et al (2014) proved that these dimensions are independent. However, research by Beese (2016) showed that structural complexity plays a very important role in dynamic complexity. Schneider, Zec, and Matthes (2014) observed that a lot of research has been conducted in the field of complexity, but a lacking dimension is a subjective category. Iacob, Monteban and van Sinderen (2018) use the work from Schneider et al. to also conceptualize subjective complexity. Their findings show that objective complexity has a major influence on subjective complexity.

The four dimensions of Schneider, Zec, and Matthes (2014) are adopted in this research. A description of the dimensions can be found in Chapter 8. Rather than a specific definition, complexity will be considered as:

Definition: Complexity: a property with a measurable value based on metrics that are relevant for the aspect under consideration.

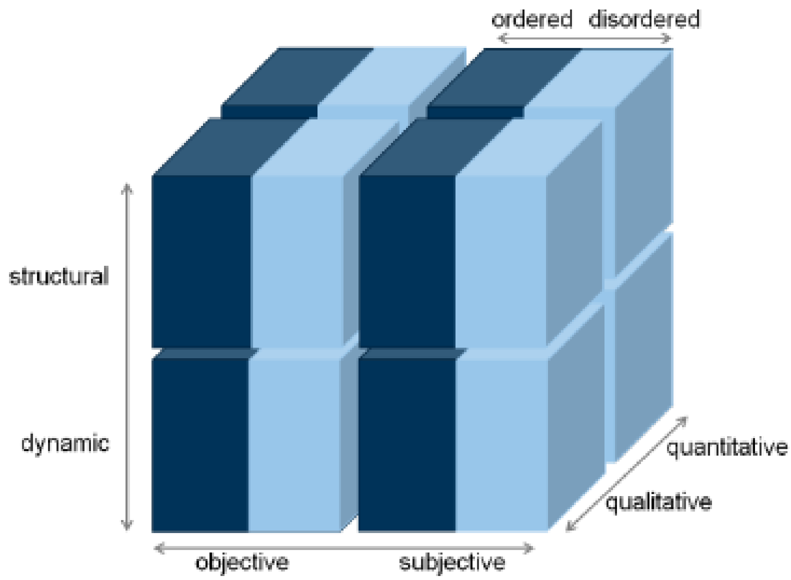


Figure 7. Complexity Dimensions (Schneider et. Al, 2014)

2.4.3 Findings on RQ3

This section describes the findings of this literature review on the third research question: What Project Portfolio Selection Method is most suitable for an extension with Enterprise Architecture Complexity?

Section 2.4.1 ended in the conclusion that Multiple Criteria Decision Analysis methods are most suitable for complex decision making, such as deciding upon which project should be selected during the Project Portfolio Selection process. Therefore, the next step is analyzing what MCDA methods are available. Ishizaka and Nemery (2013) researched the available MCDA methods and sum up the available techniques comprehensively.

The selection of an MCDA method requires some knowledge on what it requires, and of course the desired output (Ishizaka and Nemery, 2013). When the utility function for each criterion is known, Multi-Attribute Utility Theory (MAUT) is recommended. It is however a lot of work to construct such a utility function. Pairwise comparison simplifies this by comparing the criteria and their options. AHP and MACBETH are such methods. The difference between them is that AHP evaluates on a ration scale, but MACBETH on an interval scale. Outranking is based on the pairwise comparison. The options are compared two by two using an outranking or preference degree. The preference or outranking degree reflects how much better one option is than another.

	Inputs	Effort input	MCDA method	Output
Ranking/choice problem	utility function	Very HIGH	**MAUT**	Complete ranking with scores
	pairwise comparisons on a ratio scale and interdependencies		**ANP**	Complete ranking with scores
	pairwise comparisons on an interval scale		**MACBETH**	Complete ranking with scores
	pairwise comparisons on a ratio scale		**AHP**	Complete ranking with scores
	indifference, preference and veto thresholds		**ELECTRE**	Partial and complete ranking (pairwise outranking degrees)
	indifference and preference thresholds		**PROMETHEE**	Partial and complete ranking (pairwise preference degrees and scores)
	ideal option and constraints		**Goal programming**	Feasible solution with deviation score
	ideal and anti-ideal option		**TOPSIS**	Complete ranking with closeness score
	no subjective inputs required	Very LOW	**DEA**	Partial ranking with effectiveness score

Fig 8. Summary of MCDA Methods (Ishizaka & Nemery, 2013)

As the main focus of this research is the Project Portfolio Selection Process, we will select one method that will be used in the method development in Section 3.

The first option, MAUT, requires a large amount of (subjective) input. On the other hand, DEA requires no subjective input at all. The researcher hypothesizes that to calculate an overall complexity score, certain complexity metrics weigh higher than others in certain scenarios, due to a different strategic focus, or due to the preference of the user himself. To be able to distinguish those weights, a certain amount of subjective input is required. The most suitable method, in this case, would be a pairwise comparison method. In the paper of Aldea et Al (2019), the AHP method was preferred and used for their capability-based analysis method. This research agrees with the assumptions made in that research. However, research criticizes the method for its rank reversal (Ishizaka & Labib, 2011), which makes the method less reliable. Rezaei (2013) proposes an alternative method that also uses pairwise comparisons and is comparable to AHP, but has higher reliability due to it leading to more consistent ratios. This method is called Best-Worst Method (BWM). The next section will explain in more detail how the BWM works. The BWM is chosen for our method design in Section 3.

2.4.3.1 Best-Worst Method Theory

According to BWM, the best (e.g. most desirable, most important) and the worst (e.g. least desirable, least important) criteria are identified first by the decision-maker. Pairwise comparisons are then conducted between each of these two criteria (best and worst) and the other criteria. A maximin problem is then formulated and solved to determine the weights of different criteria. The weights of the alternatives for different criteria are obtained using the same process. The final scores of the alternatives are derived by aggregating the weights from different sets of criteria and alternatives, based on which the best alternative is selected. A consistency ratio is proposed for the BWM to check the reliability of the comparisons. (Rezaei, 2013)

The method consists of five steps that are used to calculate the weights of the criteria.

The first step is determining a set of decision criteria. For instance, in the case of buying a car, the decision criteria can be quality, price, comfort, safety, and style.

The second step is determining the best (e.g. most desirable, most important) and the worst (e.g. least desirable, least important) criteria. The decision-maker identifies the best and the worst criteria in general. No comparison is made at this stage. For example, for a specific decision-maker, price and style may be the best and the worst criteria, respectively.

The third step is determining the preference of the best criterion over all the other criteria using a number between 1 and 9. The resulting Best-to-Others vector would be: $A_B = (a_{B1}, a_{B2}, \dots, a_{Bn})$ where a_{Bj} indicates the preference of the best criterion B over criterion j. It is clear that $a_{BB} = 1$. For our example, the vector shows the preference of price over all the other criteria.

The fourth step is determining the preference of all the criteria over the worst criterion using a number between 1 and 9. The resulting Others-to-Worst vector would be: $A_W = (a_{1W}, a_{2W}, \dots, a_{nW})^T$ where a_{jW} indicates the preference of criterion j over the worst criterion W. It is clear that $a_{WW} = 1$. For our example, the vector shows the preference of all the criteria over style.

The fifth and final step is finding the optimal weights $(w_1^*, w_2^*, \dots, w_n^*)$. The optimal weight for the criteria is the one where, for each pair of w_B=w_j and w_j=w_W, we have w_B=w_j = a_{Bj} and $w_j/w_W = a_{jW}$. To satisfy these conditions for all j, we should find a solution where the maximum absolute differences

$$\left|\frac{w_B}{w_j} - a_{Bj}\right| \text{ and } \left|\frac{w_j}{w_w} - a_{jW}\right|$$

for all j is minimized. Considering the non-negativity and sum condition for the weights, the following problem results:

$$min \max\left\{\left|\frac{w_B}{w_j} - a_{Bj}\right|, \left|\frac{w_j}{w_w} - a_{jW}\right|\right\}$$

$$\text{s.t.}$$

$$\sum_j w_j = 1$$

$$w_j \geq 0, for\ all\ j$$

This problem can be transferred to the following problem:

$$\min \xi$$

$$\text{s.t.}$$

$$\left|\frac{w_B}{w_j} - a_{Bj}\right| \leq \xi, \text{for all j}$$
$$\left|\frac{w_j}{w_w} - a_{jW}\right| \leq \xi, \text{for all j}$$
$$\sum_j w_j = 1$$
$$w_j \geq 0, for\ all\ j$$

Solving this problem, the optimal weights $(w_1^*, w_2^*, \dots, w_n^*)$ and ξ_n are obtained.

2.4.4 Findings on RQ4

This section describes the findings of this literature review on the fourth and last research question: What metrics are used to measure complexity in Enterprise Architecture?

As recognized in section 2.4.2, the complexity of Enterprise Architecture can be distinguished into four distinct dimensions (Schneider et al., 2014). Those dimensions can be used to categorize metrics. Multiple articles mention some metrics to measure complexity in Enterprise Architecture.

However, only one article did mixed-method research to find metrics to measure complexity in Enterprise Architectures that covers complexity using Schneider's Enterprise Architecture Complexity Dimensions(Schneider et al., 2014). Iacob, Monteban and van Sinderen (2018) did a structured review on available Enterprise Architecture complexity metrics in literature and evaluated it with semi-structured interviews with experts in the field of Enterprise Architecture, which resulted in a measurement model for Enterprise Architecture Complexity. The used metrics are mapped on the four dimensions as defined by Schneider et. Al. (2014). The metrics can be found in Section 9.4. This research will use the metrics for measuring complexity in Enterprise Architecture as defined by Iacob, Monteban, and van Sinderen (2018).

3 PROJECT PORTFOLIO SELECTION METHOD

This chapter describes the proposed project portfolio selection method. The method will first be introduced in section 3.1. Then, detailed explanations are provided in section 3.2 on the various steps included in the method. The result of the method is discussed at the end of this chapter.

3.1 PROJECT SELECTION METHOD

The aim of this research is the development of a method that supports project selection by using Enterprise Architecture Complexity as a criterion in the decision process. Enterprise Architecture Complexity is recognized as an important factor in the success of a project. However, it is not recognized as a factor in project portfolio selection methods. The integration of Enterprise Architecture Complexity in the decision criteria for project selection can help organizations determine what effect a project has on the complexity of the Enterprise Architecture, which should aid the feasibility of the project and should lead to more successful projects.

The previous chapter states the importance of both Project Portfolio Management and Enterprise Architecture. In summary, Project Portfolio Management can use information from the Enterprise Architecture to determine what alternatives are most suitable for solving a specific problem. In particular, the Project Portfolio Selection process can use information about the Enterprise Architecture Complexity to quantify the impact of a project on the Enterprise Architecture. The information about the Enterprise Architecture Complexity can be useful for the Project Portfolio Management domain as it also helps to determine what projects are most important to decrease the Enterprise Architecture Complexity or to model a related strategic objective.

The realization of Enterprise Architecture de-complexification is not the only important factor in prioritizing the portfolio of projects within an organization. As also indicated in the previous chapter, the cost of the project, the implementation time, the estimated monetary benefits, and the risks of the project are also recognized in literature and practice as important factors. The main objective of the proposed method is therefore to incorporate Enterprise Architecture Complexity with the other important factors in the project portfolio selection process. The expected outcome of the method is the optimal project ranking based on the above-mentioned criteria. The proposed method is modeled in Figure 9. The next section will describe in detail how each step can be executed, accompanied by relevant available techniques. For each step, a summary explains the goal, input, and output of the step. The steps are illustrated with examples from ArchiMate models. For some steps also the application from Fortes is used, namely Fortes Change Cloud. Fortes Change Cloud is a widely used Project Portfolio Management tool.

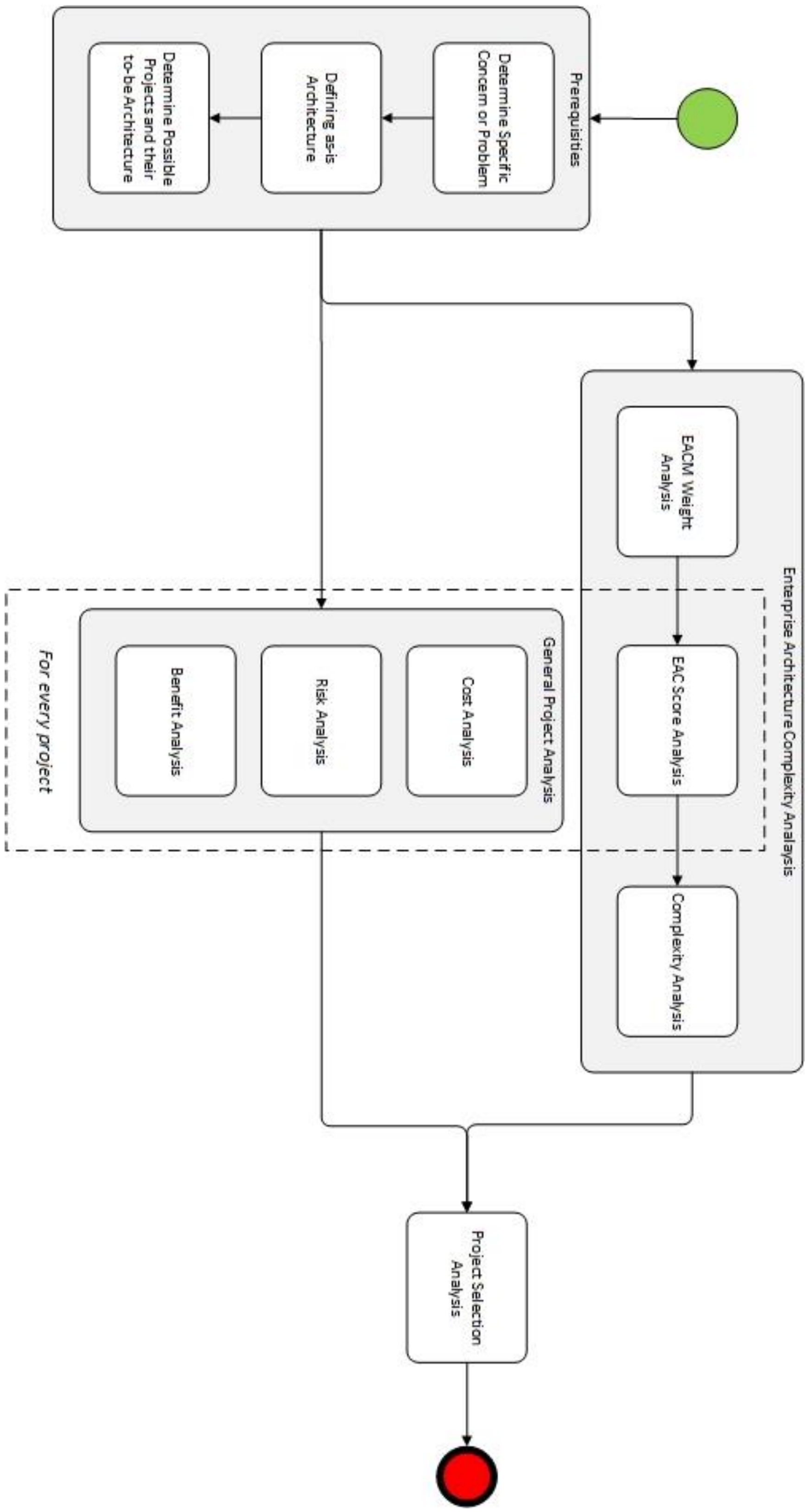


Figure 9. Proposed Project Portfolio Selection Method Design with Enterprise Architecture Complexity

3.2 METHOD DESIGN

This section describes the model from Figure 9 that is introduced in section 3.1. The model contains three different stages. First, the prerequisites are discussed in section 3.2.1. The first stage is the Enterprise Architecture Complexity Analysis and is discussed in section 3.2.2. The second stage comprises the General Project Analysis and is discussed in section 3.2.3. The third and last stage discusses the Project Selection Analysis in section 3.2.4.

Every stage in the method requires several inputs and results in different outputs. Figure 10 presents the different deliverables and their relations to the stages in the method. The prerequisites produce the as-is enterprise architecture, a project information document, and a to-be architecture for each identified project. The as-is enterprise architecture and the to-be enterprise architectures are required in the Enterprise Architecture Complexity analysis stage and produce the Complexity Analysis Results. The General Project Analysis uses all outputs of the Prerequisites and results in the Cost Analysis Results, Risk Analysis Results, and Benefit Analysis Results. The Project Selection Analysis uses the Project Information Documents from the prerequisites, the Complexity Analysis Results from the Enterprise Architecture Complexity Analysis, and the Cost Analysis Results, Risk Analysis Results, and Benefit Analysis Results from the General Project Analysis.

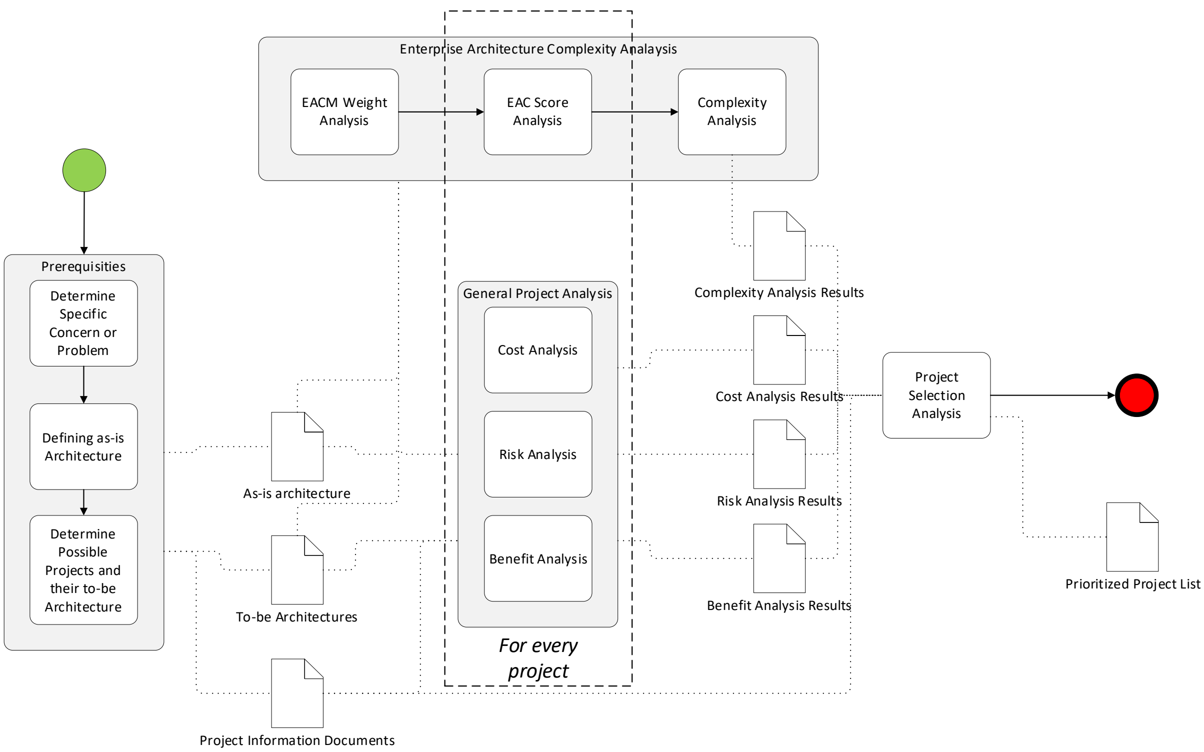


Figure 10. Proposed Project Portfolio Selection Method Design with Enterprise Architecture Complexity, with inputs and outputs presented between the stages. *The green circle indicates the start of the model, the red circle indicates the end of the model.*

The steps in the stages of the model also relate to each other. The Prerequisites and the Enterprise Architecture Complexity Analysis are using the output of the previous step in the stage as an input for the next stage. How these inputs and outputs relate to each other will be further explained in the next sections.

3.2.1 Prerequisites

The focus of this research is on the selection process, rather than determining what alternatives are suitable to solve a specific problem. Therefore, determining the problem or concern that needs to be solved is considered a prerequisite in the method. The problem should be known before applying the method. Also, the possible solutions should be defined before applying the method. The possible projects need to be determined before the application. Their appropriate to-be enterprise architectures should also be developed before the method is applied. The prerequisites are used in the next steps of the method. Figure 11 represents a schematical overview of the steps and their deliverables.

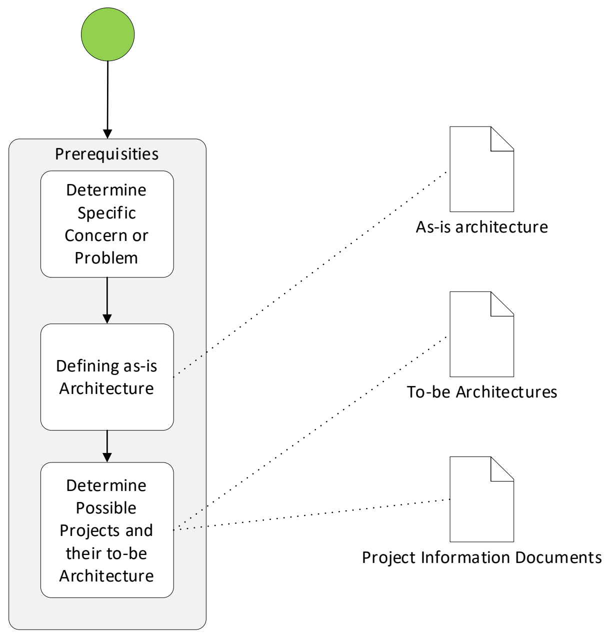


Figure 11. A schematical overview of the Prerequisites steps and their deliverables

The first stage, the Enterprise Architecture Complexity Analysis, uses information from the Enterprise Architecture of the organization. The as-is architecture describes the current situation before applying any of the possible projects. Every alternative project will have a different effect on the to-be Enterprise Architecture, and therefore possibly another effect on the Enterprise Architecture Complexity. An impact analysis, as described by Aldea et al. (2019) can be used to identify the changes in the architecture. The current as-is architecture and the possible to-be architectures for all alternative projects are considered as prerequisite knowledge. The prerequisite enterprise architectures imply that the method is only suitable for organizations that have an accurate, up-to-date enterprise architecture at their disposal.

3.2.2 Enterprise Architecture Complexity Analysis

The first stage of the method focuses on the analysis of the Enterprise Architecture Complexity. This stage can be executed when the prerequisites are fulfilled. The goal of this stage is to operationalize and measure the complexity of the Enterprise Architecture such that it can be used as a criterion for the Project Portfolio Selection Process. To measure complexity, the construct of Enterprise Architecture Complexity should be operationalized. The next section, Section 3.2.2.1 shows the operationalization of Enterprise Architecture Complexity. The operationalization is used as input for the method that is selected in Chapter 2. The Enterprise Architecture Complexity Analysis stage contains three steps: The first step is the EACM Weight analysis which is described in Section 3.2.2.2. It is followed by the Complexity Score Analysis, which should be performed for all the available alternative projects and is explained in Section 3.2.2.3. In Section 3.2.2.4, the results of the EACM Weight Analysis and the EAC Score Analysis are used for the Complexity Analysis. Figure 12 presents the inputs, the steps of the method with their appropriate input and output, resulting in the deliverable of this stage: the Complexity Analysis Results.

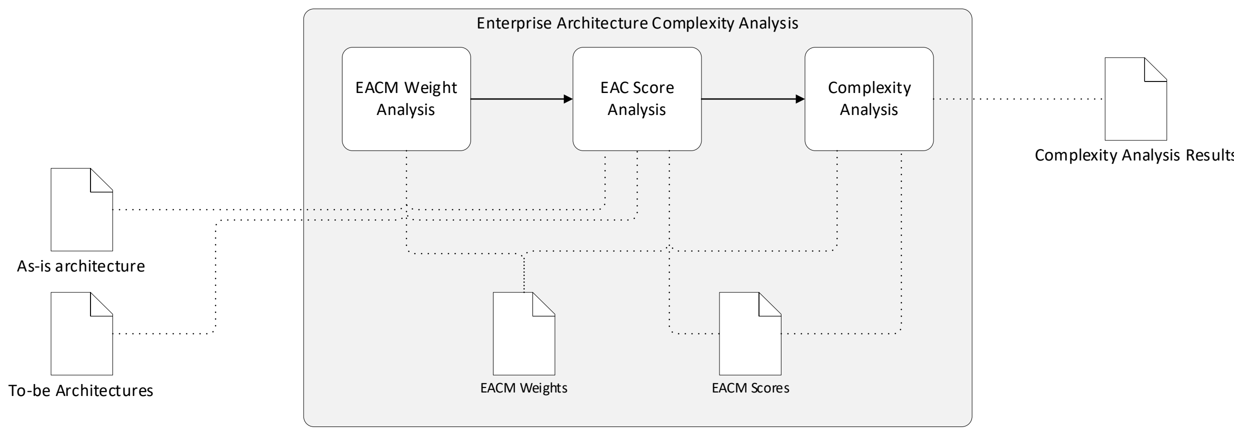


Figure 12. Enterprise Architecture Complexity Analysis

3.2.2.1 Operationalization

For the operationalization of Enterprise Architecture Complexity, we will break it down into several constructs. Lots of research is executed in this area. Schneider et al. (2014) created a model describing the different dimensions of Enterprise Architecture Complexity. Every metric involving Enterprise Architecture Complexity can be defined along the four dimensions. Monteban et al. (2018) researched complexity in this context and created a conceptual measurement model. This model is presented in Figure 13.

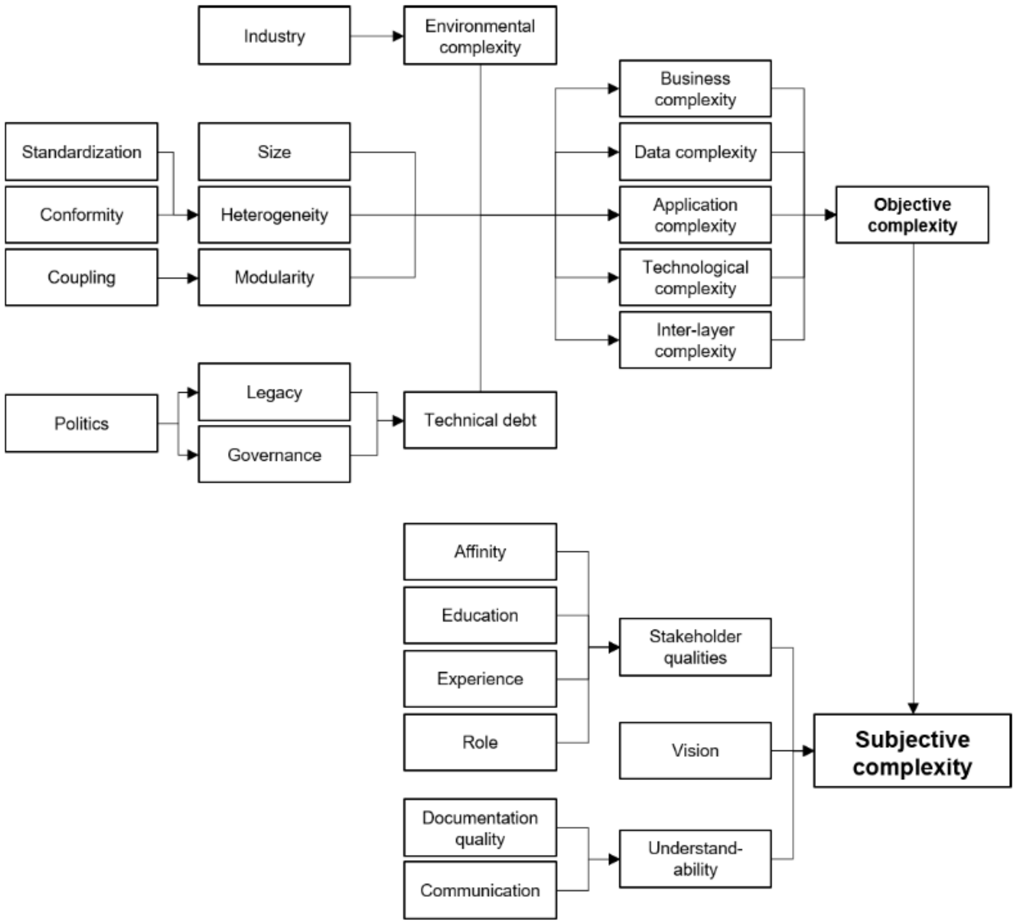


Figure 13. Conceptual model of Architectural Complexity (Monteban et. Al, 2018)

As also described in Chapter 2, the metrics that are found in the SLR by Monteban et al. (2018) are used in this research. The most prevalent metrics are filtered out, as this is an indicator of their importance. Enterprise Architecture Complexity is defined by several different constructs. Table 3 shows the constructs and their according metrics as defined by Monteban et al. (2018).

The table includes twelve constructs and twenty-eight metrics. However, we do not consider all of the metrics equally important. The subjective complexity metrics are measuring the perceptive complexity of the Enterprise Architecture. These perceptions are however not related directly to the elements of the enterprise architecture. Rather, proven by Monteban et al. (2018), the subjective complexity is influenced by the education and experience of the person, their role, and the persons' affinity with technology. Also, the level of documentation and the vision of the company, and the architecture is of influence. Important to note here is that a project is considered to result in a change in the Enterprise Architecture; "physically" in the elements and their relations itself. The level of documentation is considered to be consistent for every project, so no change will be expected in that area, making those metrics noninfluential. In regards to the vision; we think that projects are started from a certain vision, rather than a project to change the vision of the company. If this is the case, this proposed method is not suitable.

Construct	Metrics
Size	Number of elements Number of relations
Heterogeneity	Element entropy Relation entropy
Modularity	Element modularity
Technical debt	Cost of rework
Environmental complexity	Herfindahl-Hirschman index Size diversity Heterogeneity of output Specialization rate Labor diversity Asset size Capital intensity The technical level of the workforce
Mission	Company vision Architecture vision
Document quality	Available level of detail Available notational elements
Communication	Documentation availability Documentation detail suitability Documentation notation suitability Documentation consistency
Affinity	Technology affinity
Education	Education area Education level
Experience	Experience in organization Experience with enterprise architecture
Role	Role

Table 3. List of constructs and their metrics (Monteban et al., 2018). *The underlined metrics are included in the measurement model.*

Environmental complexity is a formative construct that is influenced heavily by the external environment of the organization. The external environment can have a significant effect on the complexity of an organization. However, we argue that the effect of a single project that is under consideration in this model will not have such a significant effect on the external environment that it will result in a significant effect on the complexity of the Enterprise Architecture. Therefore the environmental complexity construct is also disregarded in the measurement model for Enterprise Architecture Complexity. Due to time constraints, this assumption is not tested any further.

The underlined constructs and their metrics are considered for the measurement of Enterprise Architecture Complexity in the proposed method. These metrics are classified as objective complexity metrics, following the definitions of Schneider et al. (2014). The constructs' size, heterogeneity, and modularity are classified in the "architecture"-group. This group consists of those attributes directly relating to the elements and relations in enterprise architecture. As defined in chapter 2, an enterprise architecture consists of elements and their relations, categorized in different architectural domains (business, data, application, and technology). Each architectural domain consists of elements and relations, although these elements differ for each domain. The codes in this group are directly related to the characteristics of these elements and relations. An often mentioned metaphor, especially by the producer stakeholders, was the “spaghetti versus lasagna” comparison. An important goal of enterprise architecture is to take the spaghetti that is the IT landscape – a complex and chaotic whole of intertwined elements – and create a lasagna: structured and organized in different layers. The construct “Technical debt” is a formative construct that is influenced by governance and legacy. It is grouped by Monteban et Al (2018) in the Enterprise related constructs. Its operationalization relies on the metrics by Nord, Ozkaya, Kruchten & Gonzalez-rojas (2012), initially aimed at software architecture, which can also be translated to enterprise-level. They define the cost of architecture to be the sum of the cost of implementation and the cost of rework. Here, the cost of rework represents technical debt: it quantifies the future work that needs to be done on the architectural elements.

The next section describes the constructs and their metrics in more detail. Important to note is that, following the conceptual model in Figure 13, not all constructs are operationalized. For the sake of time and simplicity, this model is limited to the operationalized constructs that are at the end of the causal chain in the conceptual model.

Size

Before we dive deeper into the size of enterprise architecture, we have to define the definition of architecture. Mathematically speaking, an architecture can be described as $S = (T, R)$

Where S is the system, or in other words the architecture or one of its domains, T is the set of elements in that system, and R the relations between those elements (Schütz et al., 2013). The size of an Enterprise Architecture can be measured by one of the following metrics.

<u>Metric 1: Number of elements</u>

Count the total number of elements

Calculation: |T|

<u>Metric 2: Number of relations</u>

Count the number of relations

Calculation: |R|

Heterogeneity

Also, heterogeneity is operationalized in several papers. The definition of Schütz et al., (2013) is adopted in this research. Heterogeneity can be measured with the following metrics.

<u>Metric 3: Element Entropy</u>

Measures the heterogeneity of the set of elements in the respective architectural domain

Calculation:

$-\sum_{i=1}^{n} p_i \ln(p_i)$ where
$i \in T$
$p_i = relative\ frequency\ of\ element\ i$

Metric 4: Relation Entropy

Measures the heterogeneity of the set of relations in the respective architectural domain

Calculation:

$-\sum_{i=1}^{n} p_i \ln(p_i)$ where
$i \epsilon R$
$p_i = relative\ frequency\ of\ relation\ i$

Modularity

Modularity refers to the degree to which a system can be divided into separate components. The measurement of modularity in architecture has not yet been specifically studied. However, Newman (2006) defined a metric for modularity in networks that has been generally accepted. This metric focuses on networks consisting of nodes and edges, which is exactly how architecture was defined earlier. The metric is based on statistics: modularity exists when the number of edges between groups is significantly less than can be expected by chance. In this case, a lower level of modularity results in less complexity.

Metric 5: Element Modularity

Measures the modularity of a network of elements

Calculation:

$\frac{1}{4m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})s_i s_j$ where
$m\ \ = |R|$
$A_{ij} = number\ of\ edges\ between\ element\ i\ and\ j$
$k_i\ \ = degree\ of\ element\ i$
$s_i s_j = 1\ if\ i\ and\ j\ are\ in\ the\ same\ group\ and\ 0\ otherwise$

Cost of rework

The addition of complexity to information systems is theorized as mainly caused by legacy, politics, and the governance of information systems and results in technical debt (Monteban et al., 2018). Measuring politics and governance directly is difficult. The end of the causal chain is technical debt, therefore politics and governance are measured through technical debt. No metrics exist yet for measuring technical debt on an architectural level. However, metrics aimed at software architecture can be translated to the enterprise level. Nord, Ozkaya, Kruchten & Gonzalez-rojas (2012) define the cost of architecture as the cost of implementation and the cost of rework.

Metric 6: Cost of rework

Measures the cost of rework (Cr) that needs to be done on elements in the architecture

Calculation:
$\sum_k Cr(E_k)$ for all new element E_k where
$Cr(E_k) = \sum_j Cr(E_j)$
$Cr(E_j) = D(E_j, E_k) * Ci(E_j) * Pc(n-1)\ where\ D(a,b)$ is the number of dependencies between a and b, Ci is the implementation cost and $Pc(n-1)$ is the propagation cost of release $n-1$. The calculation of propagation cost is described by Baldwin, Maccormack, and Rusnak (2014).

3.2.2.2 Enterprise Architecture Complexity Metrics Weight Analysis

This section describes how the weight of the identified metrics from the previous section can be determined. As described in Chapter Two, the Best-Worst Method will be used for this analysis. This method consists of five steps that are reviewed in Section 2.4.3.1. The numbers and weights in this section are dummy data.

Step 1: Determine a set of decision criteria

The goal of this step is to determine the set of decision criteria that should be used. The operationalization of the previous section will be used as decision criteria. So, we have: { number_of_elements(c1), number_of_relations(c2), element_entropy(c3), relation_entropy(c4), element_modularity(c5), cost_of_rework(c6)}.

Step 2: Determine the best and the worst criteria

The goal of this step is to determine the best (most desirable, most important) and the worst (least desirable, least important) criteria. For example, the cost_of_rework is considered to be most important, along with the element_modularity, while the number_of_elements is considered as the least important.

Step 3: Determine the preference of the best over all other criteria

The goal of this step is to determine the weights of the best criteria over all other criteria with a number between 1-9, where 1 is equal, and 9 is the high preference of the best over the other. In the case that all criteria are weighted equally, the scores would all be 1 in the best-to-others matrix as presented in Table 4. Higher numbers indicate a higher preference of the best over the other criteria. In our example, we preferred cost_of_rework(c6) and element_modularity(c5) equally. We can choose one of the two arbitrarily as best. The result is called the Best-to-Others matrix and can be found in Table 4. The table presents dummy data

	C6	C5	C4	C3	C2	C1
Cost_of_rework	1	1	5	5	7	9

Table 4. Best-to-Others matrix with dummy data

Step 4: Determine the preference of all criteria over the worst

The goal of this step is to determine the weights of all criteria over the worst criteria with a number between 1-9, where 1 is equal and 9 is a high preference over the worst. . In the case that all criteria are weighted equally, the scores would all be 1 in the others-to-worst matrix as presented in Table 5. Higher numbers indicate a higher preference of the other over the worst criteria. In our example, Number_of_elements is considered the worst. The result is called the Others-to-Worst matrix and can be found in Table 5.

	Number_of_elements
C6	9
C5	8
C4	5
C3	5
C2	3
C1	1

Table 5. Others-to-Worst matrix with dummy data

Step 5: Determine the optimal weights

The goal of this step is to determine the optimal weights of the criteria. The calculation that is provided in Chapter 2 is quite extensive and shows how it is done mathematically. The application of this method in that same depth is considered out of the scope, as we are mostly interested in the application in the proposed method. Therefore, a Microsoft Excel-based solver program is used. This solver program can be provided when this is requested. The solver gave the following result based on the input from the examples in the previous steps. The result is presented in Table 6.

Criteria	**Weights**
number_of_elements	0,034
number_of_relations	0,062
element_entropy	0,086
relation_entropy	0,086
element_modularity	0,350
cost_of_rework	0,383
Ksi*	0,081

Table 6. BWM Solver result with dummy data

The Ksi* indicates that the solution is consistent. For more information on how to optimize the consistency of the solution, we refer to Rezaei (2014).

The determination of the weights can also be considered as a collaborative effort among several decision-makers. To successfully determine the weights, this step can be executed for each individual. The resulting weights are then analyzed. The resulting weights are the means of the respective criteria.

3.2.2.3 Enterprise Architecture Complexity Score Analysis

This section describes how the metrics that are identified in the operationalization in Section 3.2.2.1 should be measured. This step should be applied to every project that is considered in the project selection process.

The to-be architecture and the available alternative project's to-be architecture needs to be scored according to the defined operationalization. The metrics are applied to the architecture, resulting in a table with a column for each project. These scores will be used in the Complexity Analysis. The result could look like Table 7. The data in the table is dummy data.

Metric	**As-is**	**Score P1**	**Score P2**	**Score P3**	**Score P4**
number_of_elements	34	30	28	32	38
number_of_relations	34	30	28	32	38
element_entropy	0.7	1	0.8	0.5	0.8
relation_entropy	0.7	1	0.8	0.7	0.5
element_modularity	0.5	0.6	0.4	0.7	0.3
cost_of_rework	12	10	8	12	11

Table 7. Result of Enterprise Architecture Complexity Score Analysis

3.2.2.4 Complexity Analysis

The previous sections described the operationalization of Enterprise Architecture Complexity, how the Best-Worst Method can be applied to determine the weights of the metrics, and the last section determined the scores for all alternative projects. This section will describe how the results from the previous sections conclude in a single complexity number.

Al-Harbi (2001) researched the use of AHP-like methods for project selection, and how to use the results of an MCDA. The method of Russel et al. (1987), the dimensional weighting method is used to determine the score. In this method, the choice selection criteria and their weights are dependent on the owner. All alternatives are ranked based on the criteria. An alternative's total score is calculated by summing their ranks multiplied by the weight of the respective criteria. Then, the alternatives are ranked based on their total scores.

The Complexity analysis stage is applied for every project that is considered in the project selection process. However, the calculation of the weights only needs to happen once. The weights are based on the preference of certain criteria. These preferences can change over time, but can also be different for various aspects of an entire architecture and different parts of an organization. For every alternative project, the to-be architecture is scored according to the defined operationalization. The scores should be multiplied by the weights to determine the overall complexity score per project. Before those scores can be multiplied by the calculated weights, they should be normalized.

The magnitude of certain metrics, such as *number_of_elements*, can be very high compared to element_modularity, which is between 0 and 1. To be able to combine all six metrics into one measure, the scores will be parsed into a number between 0 and 1. The maximum value for each metric is considered as one, except for modularity of which the domain is already between 0 and 1. However, because higher modularity indicates less complexity, we subtract the modularity score from 1 to calculate the lack of modularity. The other values are then divided by the maximum value to determine the relative score.

The next step is determining the Complexity score per project. The Score of each metric is multiplied by the weight of that metric and then summed up for each project. The sum is the complexity score. The used Excel file for this calculation can be provided on request.

Metric	**As-is**	**To-be P1**	**To-be P2**	**To-be P3**	**To-be P4**	**Weights**
number_of_elements	0,895	0,789	0,737	0,842	1	**0,034**
number_of_relations	0,895	0,789	0,737	0,842	1	**0,062**
element_entropy	0.7	1	0,8	0,5	0,8	**0,086**
relation_entropy	0.7	1	0,8	0,7	0,5	**0,086**
element_modularity	0.5	0,4	0,6	0,3	0,7	**0,350**
cost_of_rework	1	0,833	0,667	1	0,917	**0,383**
Weighted Total	**0,764**	**0,706**	**0,673**	**0,671**	**0,803**	
Complexity Score		**0,057**	**0,091**	**0,092**	**-0,039**	

Table 8. Result of Complexity Analysis

This complexity score indicates the complexity based on the weights of the criteria and the score of the criteria. Therefore, the complexity score can be different for the same architecture when a Metric is prioritized differently in the EACM Weight Analysis. The output of this stage, the complexity score for each project, serves as input for the Project Selection Analysis in Section 3.2.4.

3.2.3 General Project Analysis

The second stage of the proposed Project Portfolio Selection Process describes the General Project Analysis. In this stage, three common types of analysis are applied for project selection. Cost, risk, and benefit are analyzed in Section 3.2.3.1, Section 3.2.3.2, Section 3.2.3.3, respectively. As these types of analysis are widely studied and a variety of methods is used in practice, this research is limited to the methods that can be used in relation to the Enterprise Architecture models from the previous steps. Therefore, we will not go into a lot of detail in regards to the application of the analysis. References are provided for further information on the type of analysis.

The outputs of the Prerequisites: the as-is architecture, to-be architectures, and the project information documents are used as input for the General Project Analysis stage. Cost, risk, and benefit analyses are applied and result in the following outputs: Cost Analysis Results, Risk Analysis Results, and Benefit Analysis Results. The schematical representation of this stage is presented in Figure 14.

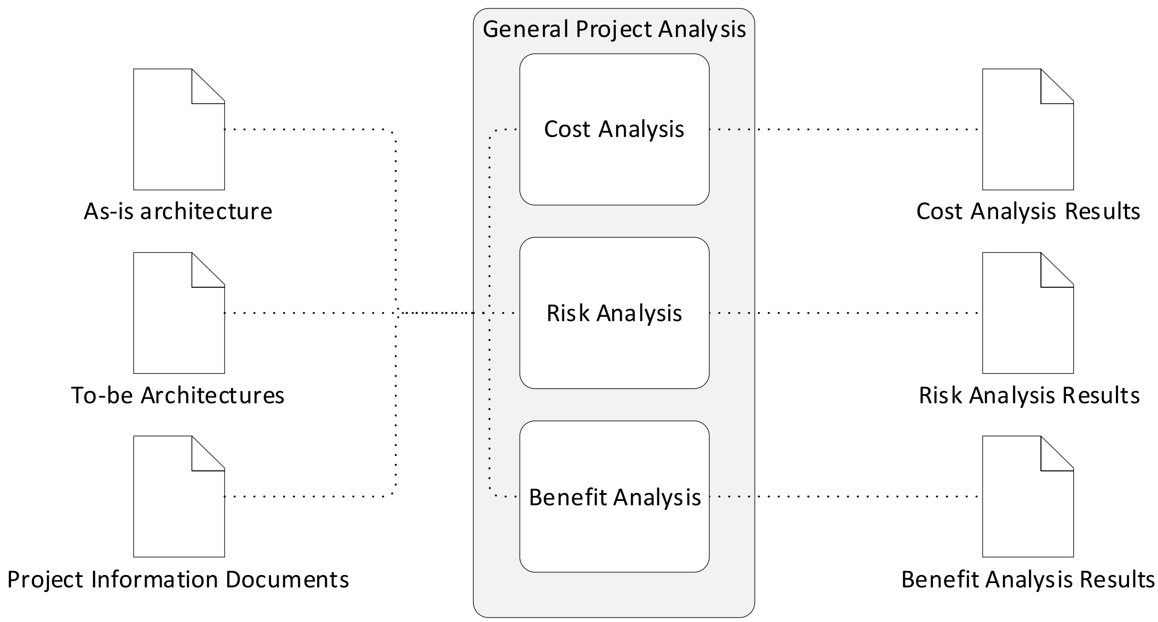


Figure 14. General Project Analysis

3.2.3.1 Cost analysis

The goal of the cost analysis is to determine the expected cost of the project. This analysis can be based on the difference between the as-is and to-be architecture, or other cost calculation methods. For this example, we will use the outputs of the prerequisites, namely the differences between the as-is and the to-be architecture, as suggested by Iacob et Al (2012). Each of the changes in the architecture should be analyzed along with the associated costs. If there is no change on a specific element in the architecture, no costs will be analyzed for this element. There should be a deletion, modification, or addition. Figure 15 shows an example architecture with some changes. The white color indicates that no changes are needed. The grey color indicates a deletion. The yellow color indicates modification, and the red color indicates an addition. Note that also relations are colored, as these can indicate interfaces that need to be deleted, modified, or added as a result of the change.

Based on Iacob, Quartel, et al. (2012) in their paper, the cost is a property that practically can be associated with any architectural entity and/or a specific project. Above the elements and relations is room for the specification of the type of cost required to realize the proposed change. The sum of all those changes is considered as the total cost of the project and will be used in the Project Selection Analysis in Section 3.2.4.

As also described in Iacob et Al (2012), costs do not need to be defined in monetary values directly. In the example below also man-hours are defined as an indirect way of measuring monetary value by defining the necessary effort. These values need to be translated to monetary values. In the total cost calculation, these values need to be multiplied by the cost per manhour. Note that all three types of actions involving an element in the architecture can have monetary value, as given in Figure 15.

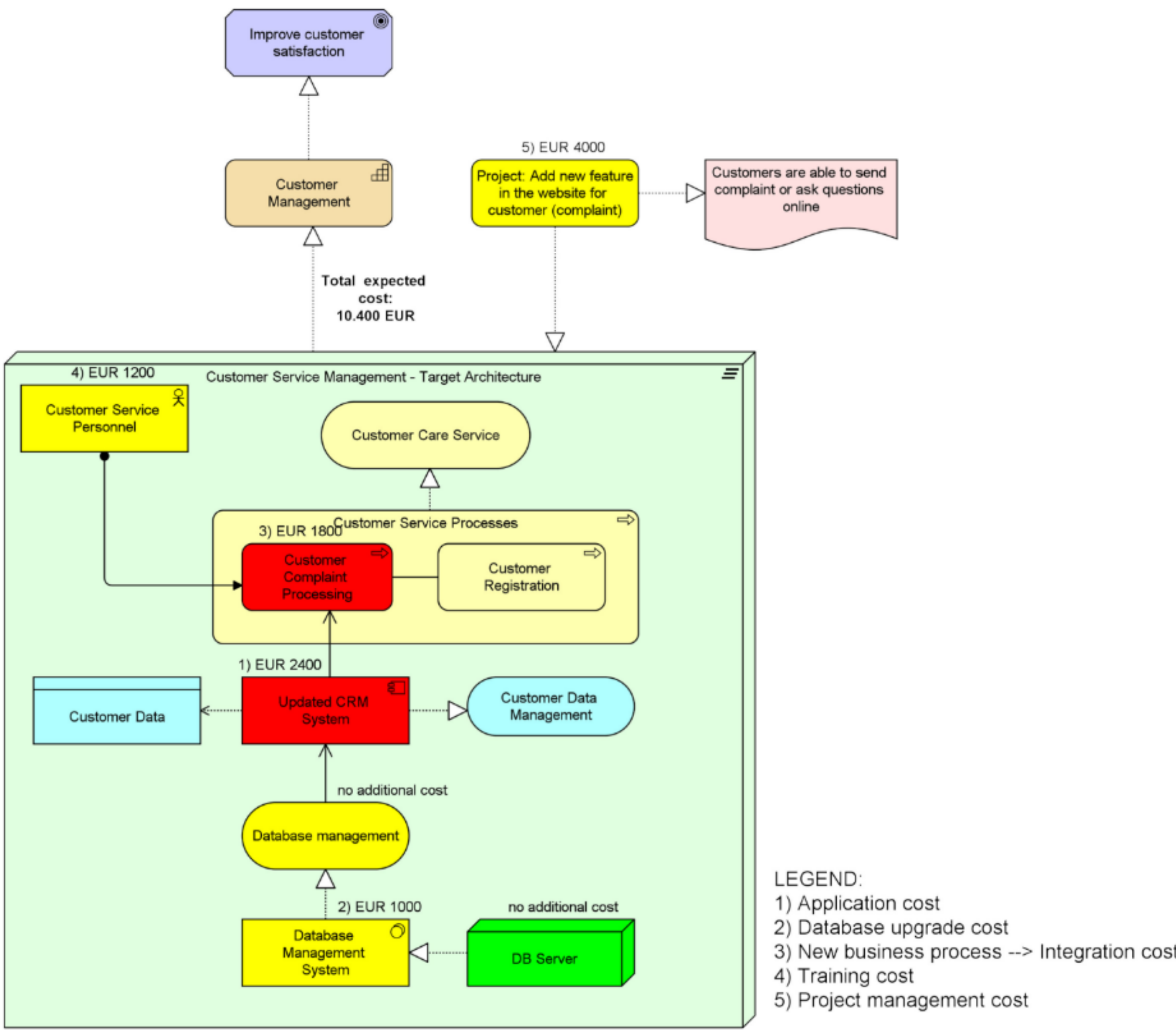


Figure 15. ArchiMate Model-Based Cost analysis (Aldea et al., 2019)

3.2.3.2 Risk Analysis

After the cost analysis, a risk analysis should be performed to determine possible risks that can occur due to the project. Certain changes in the architecture can lead to value loss. This step's goal is to identify those risks, determine the likelihood that this risk occurs, and calculate the approximate cost. The Open Group (2009) defines risk as "the probable frequency and magnitude of loss that arises from a threat (whether human, animal or natural event)".

Risk management usually includes the activities of establishing the context, assessing (identifying, analyzing, and evaluating), treating, communicating, consulting, monitoring, and reviewing the risks (Barateiro et al., 2012). The main activities in the proposed method will follow these activities partly. The risk analysis in our proposed method includes establishing the context, risk identification, and analysis. The rest of the steps are considered as part of Project Management, as those are executed after the project selection during the implementation, and therefore excluded.

First, the context. For the proposed method, the context is the risk of the possible alternative projects. The identification and analysis are done before the project execution, therefore the risk identification is based on the changes in the architecture that are supplied through the prerequisites. The first step is the determination of events from the changes in the architecture that could lead to value loss. From the events that could lead to value loss, the level of impact and the probability that the risk happens should be defined. From those two values, the risk factor can be calculated. The most common risk calculation formula is the probability that the risk happens multiplied with the magnitude of the effect, i.e., the size of the value loss. This process should be repeated for all the identified risks. When all risks for a project are identified and analyzed, a total risk score should be determined. This is the average risk score. This result will be used in the Project Selection Analysis in Section 3.2.4.

Multiple methods are available to model and calculate risk. The Open Group (2009) created guidelines on modeling risks in the ArchiMate language. The risk concept is not introduced as an independent concept, but rather as a specialization of the assessment concept from the motivation extension, since it represents the outcome of some risk assessment. It is also important to note that the basis of these concepts is the consolidation of risk and security concepts, thus it is more applicable for security risks, which is more technical compared to project risks. However, some of the concepts are still relevant to be used for modeling project risks (Aldea et al., 2019). Table 9. represents the ArchiMate Concepts used for project risk modeling.

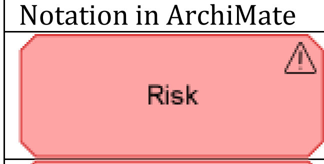
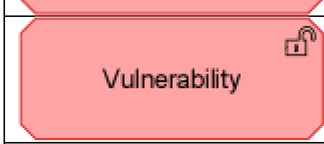
Notation in ArchiMate	Concept	Parent Concept	Description
	Risk	Assessment	The probable frequency and magnitude of future loss
	Vulnerability	Assessment	The probability that an asset will be unable to resist the actions of a threat agent
	Loss Event	Business Event	Any circumstance that causes a loss or damage to an asset

Table 9. ArchiMate Concept for Project Risk Modelling

A risk model is presented in Figure 16, by using the concepts from Table 9.

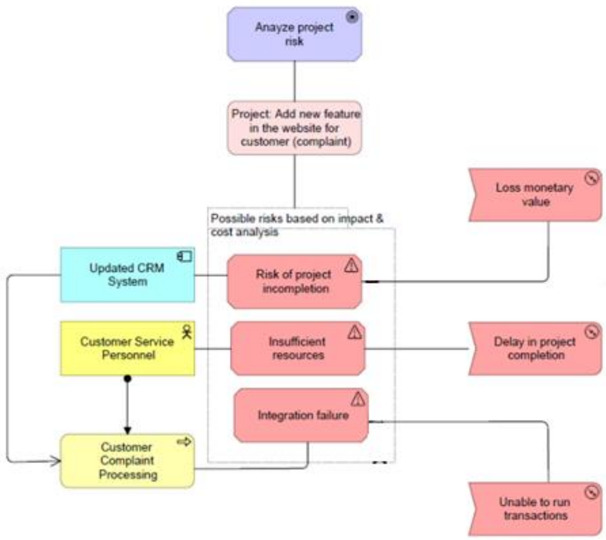


Figure 16. ArchiMate Model-Based Risk Analysis (Aldea et al., 2019)

A quantitative security risk analysis method is proposed by Breu et al. (2008). Although we need to model project risks, we can apply the quantitative security risk analysis method to project risk as well. The method is based on an Enterprise Architecture. The method extends security management methods with concepts and methods to provide the possibility for quantitative analysis. Several security metrics are introduced and it is explained how they can be aggregated by using the underlying model as a frame. For starters, you measure the number of attacks of certain threats and estimate their likelihood of propagation along the dependencies in the underlying model. Using this approach you can identify which threats have the strongest impact on business security objectives and how various security controls might differ with regard to their effect in reducing these threats (Breu et al., 2008).

Visualizing the risk can be done in multiple ways. A popular way is the Risk Heat Map. The Risk Heat Map models the impact and its probability in a table. An example is presented in Figure 17. The vertical axis measures the impact of the risk, and the horizontal axis measures the frequency/probability of the risk occurring. The color in the table represents the risk level and risk score. The heat conversion is pre-determined, and ranges from low to critical, and corresponds to a risk score ranging from 1 to 4. The final risk score can be determined by taking the average risk score of all the risks that are identified in the steps before.

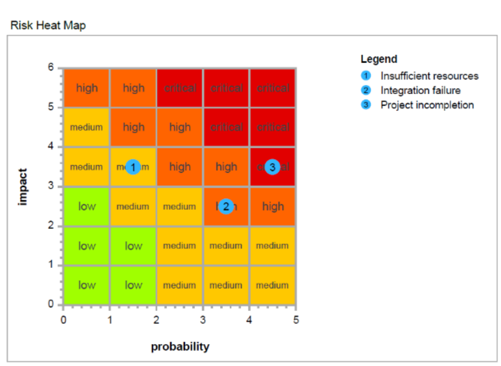


Figure 17. Risk Heat Map Example

3.2.3.3 Benefit analysis

Benefit analysis is often mentioned in one breath with cost analysis, cost-benefit analysis, or CBA. On the contrary to costs, benefits are results or outcomes of a positive nature. Important to note is that the benefits that are measured in this project selection method are in the form of an expected benefit that will be realized by the completion of a project. These techniques are also part of project valuation methods. Those financial calculation techniques try to measure project benefit as a monetary value. However, the task of estimating that monetary value is challenging, especially in the case of long-term and intangible benefits.

From an architectural viewpoint, something in the architecture triggers a value gain, which results in a Benefit. In ArchiMate, the concept of benefit is approached similarly to the risk concept. Benefits can be modeled with the Assessment element. The trigger for the benefit is modeled as an event. These events can cause a positive value gain for the organization. In ArchiMate this relationship is modeled with an influence relationship. In the case of benefits, it is a positive influence relationship. The type of event is dependent on the Architecture and to what layers it belongs. The difference between benefit and risk is the relationship: in the case of risk, it is a negative influence, while it is positive for benefits. This distinction is visualized in Table 10.

Events	Relationships	Effects
Loss event	− ---------------> Negative influence	Risk
Business event Application event Technology event Implementation event	+ ---------------> Positive influence	Benefit

Table 10. Risk and Benefit Concept Comparison

The benefits can be modeled using ArchiMate. An example benefit analysis is presented in Figure 18.

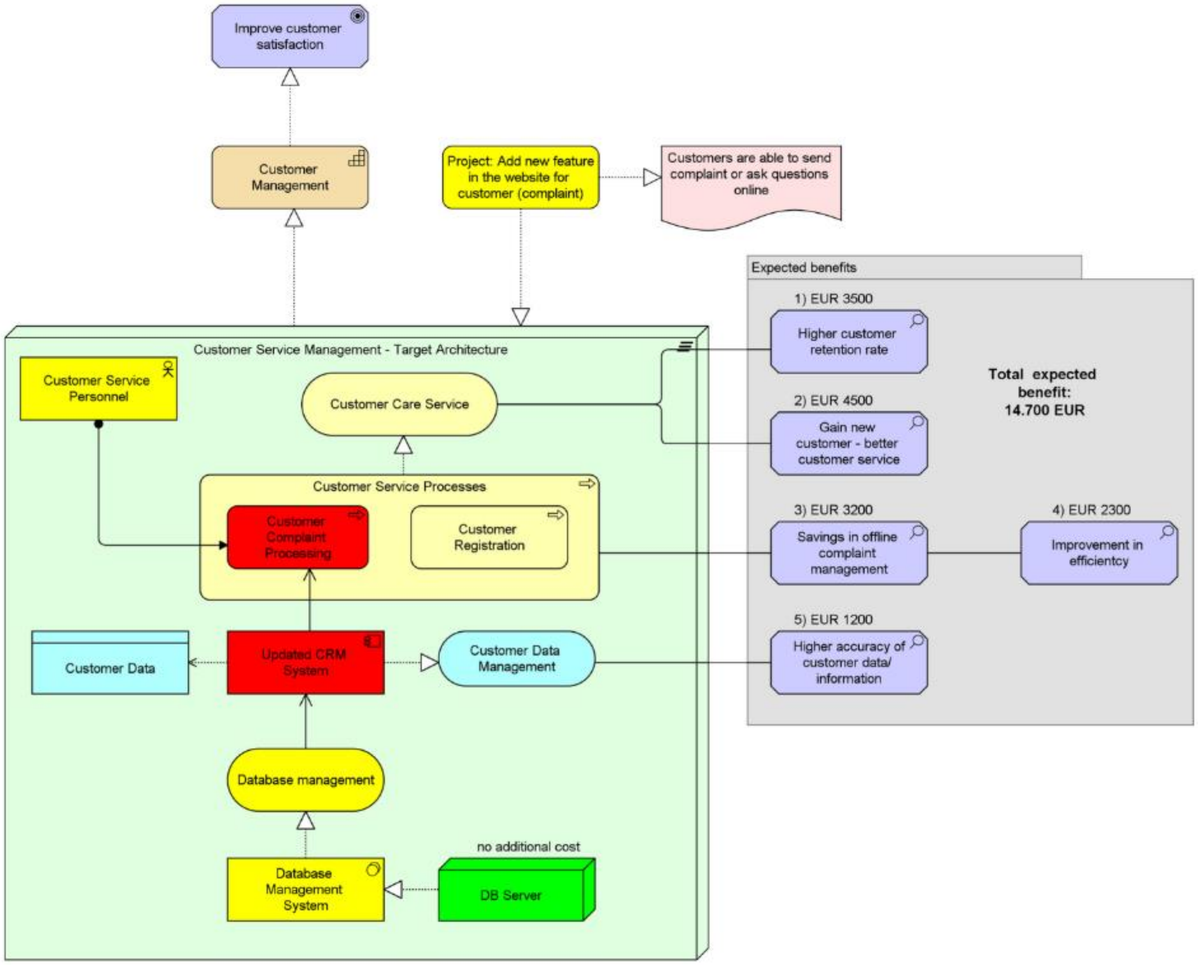


Figure 18. ArchiMate Model-Based Benefit Analysis (Aldea et al., 2019)

After the identification of the different benefits that should be achieved when implementing the project, the benefits should be valued. The example above uses monetary values to express the benefits of certain architectural changes. The sum of these benefits is the total amount of benefits for that specific project. When applying this method for project selection where the benefits are long-term or intangible, a different approach can be used, that was introduced by Aldea et al. (2019).

Score	Description
1	Very Low Importance
2	Low Importance
3	Moderate Importance
4	High Importance
5	Very High Importance

Table 11. Benefit Importance Levels

After the identification of the benefits, every benefit should be scored using the above table. The total score is than the average of the score per identified benefit. It is also possible to assign an importance score based on the identified monetary value of the benefit. Important to note is that the score should be higher when a benefit is considered better, as the goal is to maximize the benefits of a project.

3.2.4 Project Selection Analysis

The third and last stage of the proposed Project Portfolio Selection Process is the Project Selection Analysis. The goal of this step is to select the best (most suitable) project based on the provided criteria from the analyses of the previous stages. This step of the method uses the same type of Multi-Criteria Decision Analysis as defined in Section 3.2.2.2. An EA-based representation of the optimization problem is presented in Figure 19.

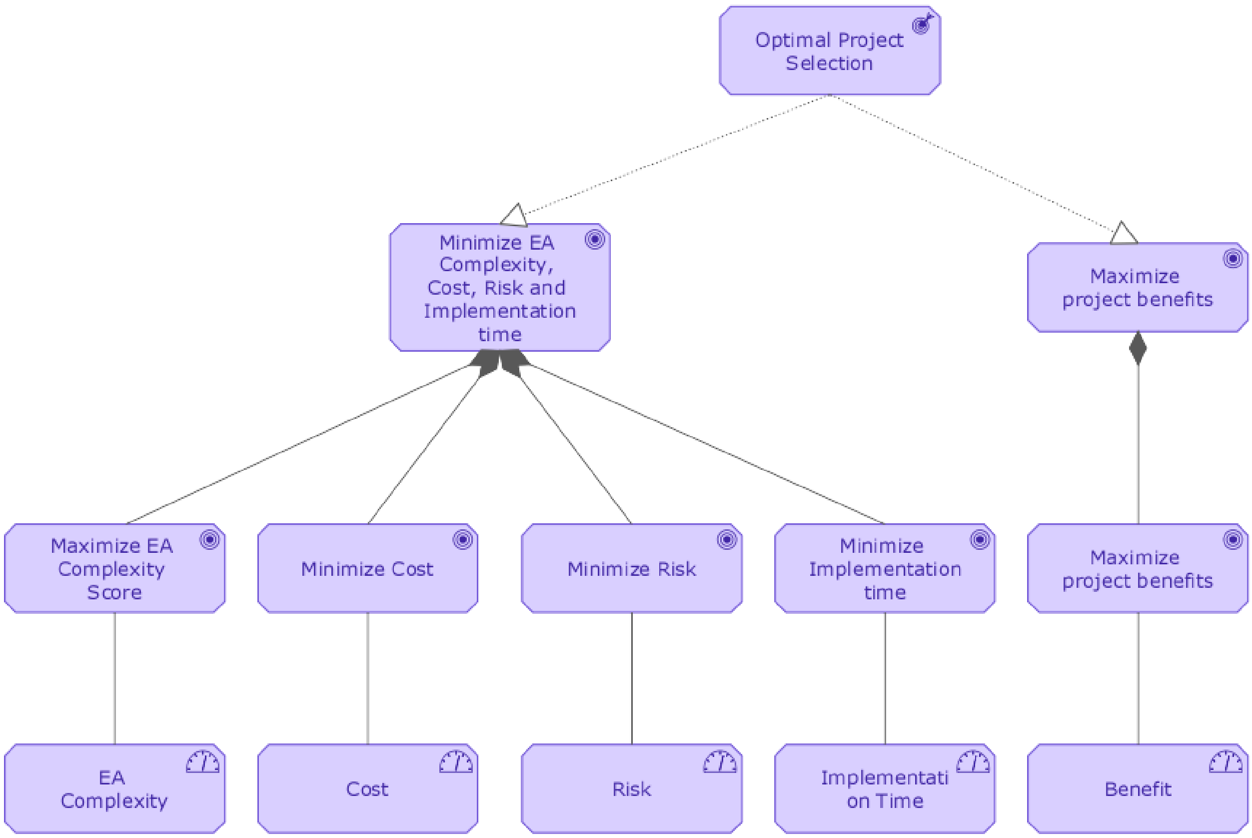


Figure 19. EA-model for Project Selection Optimization Problem

This step requires several inputs from several stages. Table 12 describes the different input criteria for the Project Selection Analysis. The Project Selection Analysis consists of a Multi-Criteria Decision Analysis, specifically the Best-Worst Method. The output of this step is a prioritized list of projects. Figure 20 presents a schematical overview of this step.

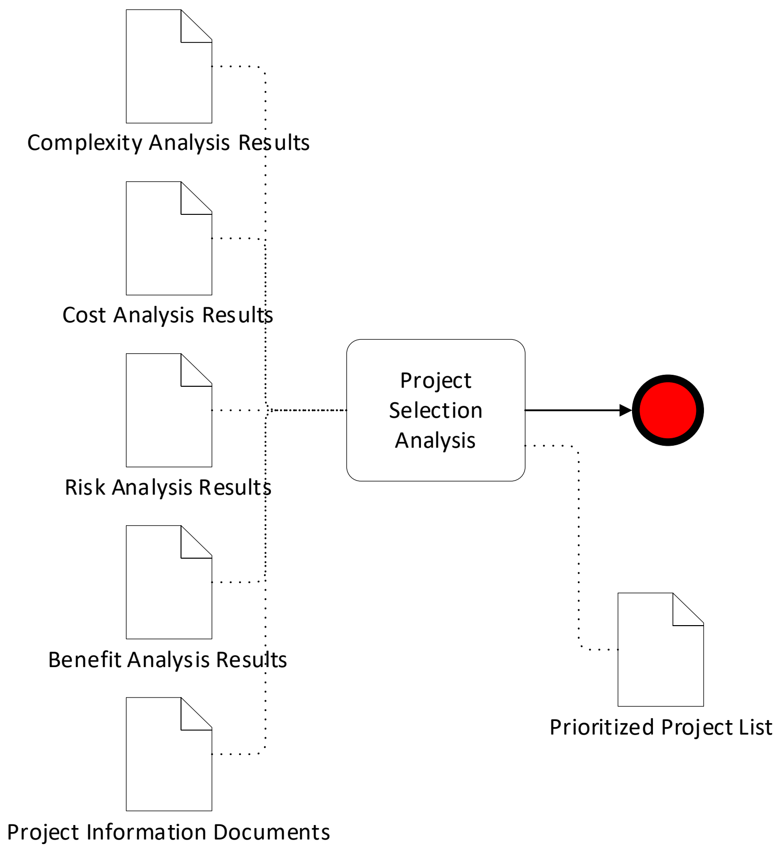


Figure 20. Project Selection Analysis

As described in Section 3.1, multiple criteria are used in the Project Selection Analysis. Table 12. describes what criteria are included, and their source.

Criteria	Source	Section
Complexity Score	Complexity Analysis Results	Section 3.2.2.4
Cost	Cost Analysis Results	Section 3.2.3.1
Risk Score	Risk Analysis Results	Section 3.2.3.2
Benefit Score	Benefit Analysis Results	Section 3.2.3.3
Project Duration	Project Information Documents	Section 3.2.1

Table 12. Project Selection Analysis Criteria

The first step of the Project Selection Analysis is combining the output of all the previous steps for all the project alternatives. For example, we used four projects in the complexity analysis. The costs, risks, benefits, and project durations should be combined for the four projects. An example can be found in Table 13. The data in the table is dummy data.

Criteria	P1	P2	P3	P4
Complexity Score	0,05709	0,090503	0,092145	-0,03937
Cost	33	24	29	44
Risk Score	3	1	2	4
Benefit Score	3	3	2	4
Project Duration	18	12	15	13

Table 13. Project Selection Criteria Scores

As described in Chapter 2 and Section 3.2.2, the Best-Worst Method is applied. The application is a little bit different from Section 3.2.2. In the Complexity Analysis, the Best-Worst Method was applied on the criteria to determine the weights. As described in [BWM], the method can be applied in two ways. In the Project Selection Analysis, the Best-Worst Method is applied to the projects per criteria. This is called the dimensional weighting method. The general description by Russel et al. (1987) is as follows: *"In the dimensional weighting method (Russell et al., 1987), the choice selection criteria and their weights are dependent on the owner. All contractors are ranked based on the criteria. A contractor's total score is calculated by summing their ranks multiplied by the weight of the respective criteria. Then, contractors are ranked based on their total scores, and this rank order of the contractors is used for prequalification."* This example is given in the context of contractor selection. However, we are interested in the most suitable project. So in this case, for the five criteria, the scores of the four project alternatives are compared. Then, after applying the Best-Worst Method five times on the different criteria, the weights of the criteria are summed up per project alternative, as described in the dimensional weighting method by Russel et al. (1987). The highest summed-up score is the best (most suitable) project.

To illustrate the other type of application of the Best-Worst Method, we will apply it on one of the criteria. The five steps of the Best-Worst Method are applied to the Complexity Score. The method should be applied for each one of the five criteria.

Step 1: Determine a set of decision criteria

The first step of the Best-Worst Method is determining the set of decision criteria. As described, the decision criteria, or the alternatives, are {P1, P2, P3, P4}.

Step 2: Determine the best and the worst criteria

The goal of this step is to determine the best (most desirable, most important) and the worst (least desirable, least important) criteria. The best and worst are in this case dependent on what criteria are under consideration. The Complexity Score should be maximized, as it indicates the contribution to the de-complexification of the architecture. The best criteria/project is in this case the project with the highest Complexity Score, which is P3. The worst alternative is in this case P4 for the Complexity Score. In the case of costs, the goal is to minimize. In that case, the smallest cost should be considered as the best alternative.

Step 3: Determine the preference of the best overall criteria

The goal of this step is to determine the weights of the best criteria over all other criteria with a number between 1-9, where 1 is equal, and 9 is the high preference of the best over the other.

	P1	**P2**	**P3**	**P4**
P3	3	2	1	7

Table 14. Best-to-Others matrix

Step 4: Determine the preference of all criteria over the worst

The goal of this step is to determine the weights of all criteria over the worst criteria with a number between 1-9, where 1 is equal and 9 is a high preference over the worst. In this example, P4 is considered the worst. The result is called the Others-to-Worst matrix and can be found in Table 15.

	P4
P1	6
P2	7
P3	8
P4	1

Table 15. Others-to-Worst matrix

Step 5: Determine the optimal weights

The goal of this step is to determine the optimal weights of the criteria. Again, the Microsoft Excel-based solver program is used. This solver program can be provided when this is requested. The solver gave the following result based on the input from the examples in the previous steps. The result can be found in Table 16. The table also shows the weights of the other criteria.

Criteria	P1 Weight	P2 Weight	P3 Weight	P4 Weight	Sum
Complexity Score	0,193	0,289	0,467	0,051	1
Cost	0,132	0,590	0,220	0,058	1
Risk Score	0,172	0,466	0,259	0,103	1
Benefit Score	0,226	0,226	0,129	0,419	1
Project Duration	0,066	0,474	0,184	0,276	1
Total	**0,789**	**2,045**	**1,259**	**0,907**	

Table 16. Weights and Sum per Project

The goal of this step is to get a prioritized list of alternatives based on the criteria. The prioritization is based on the sum of the weights of the different criteria per project. In this case, the best alternative project is P2 with a total score of 2,045. The scores for the projects can be found in Table 17.

Project	Score
P2	**2,045**
P3	**1,259**
P4	**0,907**
P1	**0,789**

Table 17. Prioritized projects

4 DEMONSTRATION

The goal of this chapter is to demonstrate the proposed method in a real case. This is part of the Design Science Research Methodology as described in Chapter 1. The application of the proposed method is particularly important because it proves that the method is applicable in real circumstances in organizations. The proposed method will be demonstrated on a case from a medium-sized software vendor in the Netherlands called Fortes Software BV, which will be referred to as Fortes. Their software package Fortes Change Cloud is used by large, international organizations and governments. Information about the case is gathered within the company.

The case of Fortes is explained first in Section 4.1, along with some assumptions that are needed for the application of the method. Next, Section 4.2 describes in detail how the proposed method can be used in a real context.

4.1 CASE DESCRIPTION

This section aims to give a complete description of the company and the context of the project. Fortes is an independent software vendor that delivers project portfolio management software to its clients. The majority of the clients use the software in the cloud, while some clients have chosen to keep the software on their servers. Fortes continuously updates their software and releases new functionalities to all clients.

The goal of Fortes is to deliver quality software to their clients that are easy to implement and use, and supports best practices such as Agile methodologies and Prince2. Fortes Change Cloud (FCC) delivers apps for strategy mapping to execution in Agile teams. With flexible connectivity applications, connections can be made with applications used by those teams such as Atlassian Jira and Microsoft Teams. Data can be analyzed with powerful connectivity towards BI analytics tools, and also supports the extraction of data inside FCC itself. Fortes develops all of its software in-house. Two development teams operate in Enschede and deliver software through Scrum methodologies. The software is implemented at customers mostly by partners, but also by Fortes on occasion. Technical and functional support is delivered by a team of specialists.

A few years ago, Fortes transformed significantly. The goal was to be more agile and get rid of existing bubbles within the organization. Adopting scrum was a step in the right direction. The teams now operate in a more integrated manner. However, the application landscape remained fragmented. All the original business silos were still using their specialized software packages. Performing extensive customer analyses was difficult as all data was fractured between those packages. But it also stagnated the initiatives for internal improvements. Changes required extensive adaptation in the interfaces between the different applications. An application consolidation initiative was started to address the abovementioned challenges.

4.2 METHOD APPLICATION

This section will describe how the proposed method can be applied to the case of Fortes that is described in the previous section. This case study addresses five project alternatives that aim to solve the problems that are described in the previous section. These alternatives are credible alternatives that are fully based on real data and information.

4.2.1 Step 0: Prerequisites

This section describes the prerequisites of the proposed method. As defined in Section 3, the method requires three things before the method can be applied. The three steps are addressed in the next sections. The level of detail in the description of these steps is limited.

4.2.1.1 Determine specific problem

The specific problems Fortes is facing are related to the recently performed agile transformation. The Customer Success Team is not working efficiently yet due to the fragmented application landscape. Every silo is using its specialized software, which makes it very difficult to combine information. This also does not benefit the customer, as the fragmented application landscape forces a lot of manual processes that make those processes more time-intensive.

4.2.1.2 Determine as-is architecture

The as-is architecture represents the current state of the enterprise architecture of the organization. The current situation at Fortes is fairly fragmented due to the different business applications in use per silo. The current situation is represented in Figure 21.

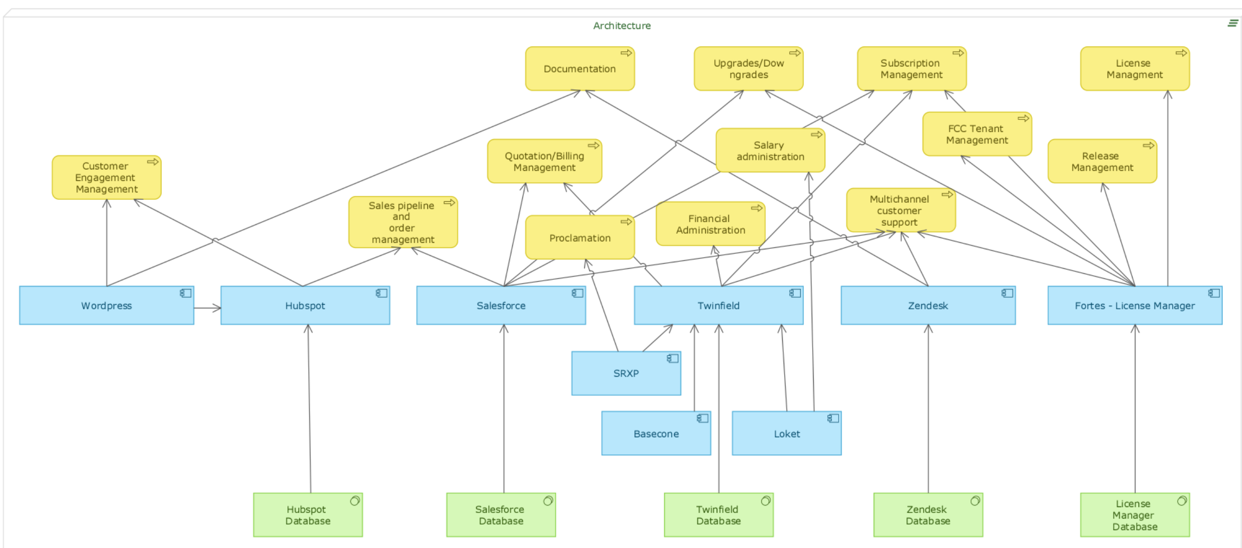


Figure 21. As-Is Architecture Fortes

4.2.1.3 Determine to-be architectures and their project information documents

Several different projects are identified by Fortes. Five are demonstrated in this case study. The starting position of Fortes is the Enterprise Architecture represented in Figure 21. All five projects propose a compelling business case and include some major changes in the architecture, specifically in the used business applications. Some restrictions in detail are put in the designs because of the otherwise very complex architectures, that do not necessarily benefit the purpose of the case study.

The next sections will introduce the alternative projects in more detail. The sections include a general description of the project, the kind of impact the project has on the architecture with appropriate assumptions. This also includes an estimated duration of the project and the corresponding to-be architectures.

Project 1

The first project proposes a radical change by migrating all business applications into one, an ERP package called Odoo. This ERP system supports all business processes within Fortes and is therefore suggested as a possible replacement for all existing software. However, replacing all existing systems involves a significant investment in both time and money. Full implementation of this ERP system is estimated to take 2 years. This project is the most drastic and requires the most effort to migrate all systems to one.

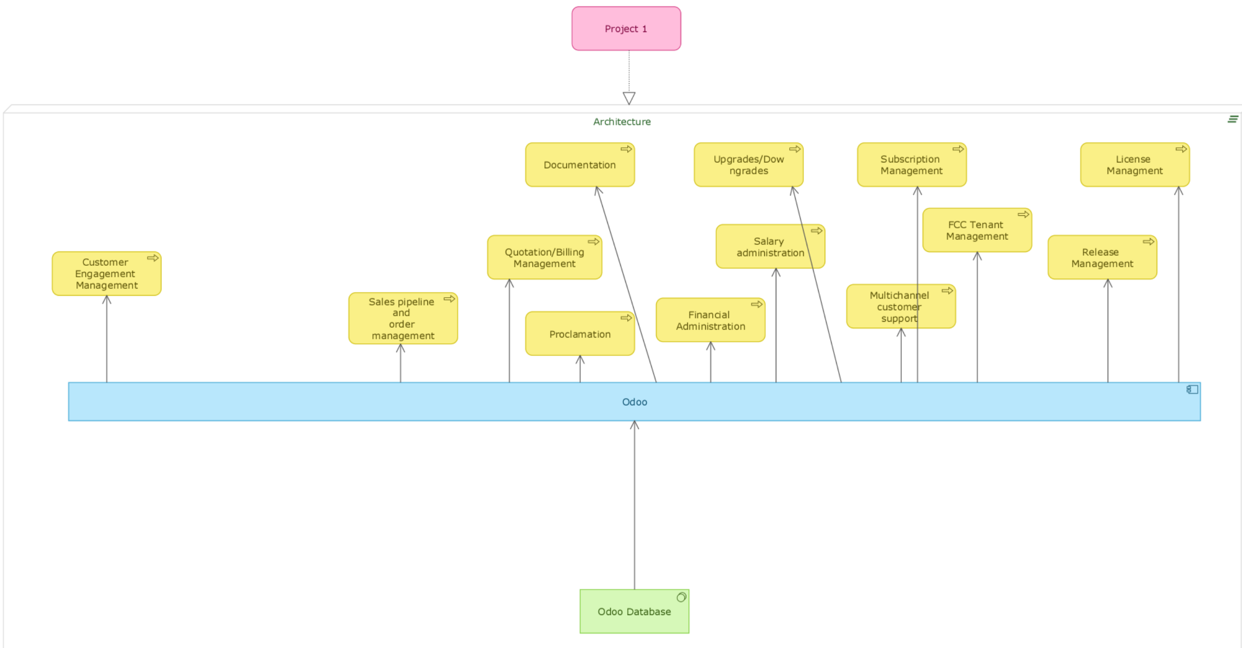


Figure 22. To-be Architecture Project 1

Project 2

The second project is less drastic than the first project. Still, a lot of applications are proposed to be migrated. However, Twinfield is still used, including already integrated applications. This project preserves financial administration. However, a relation between Odoo and Twinfield needs to be created to get the financial data from Odoo to Twinfield. The migration of the financial administration is a time-intensive task. Without it, the project implementation time is estimated to take 1,5 years.

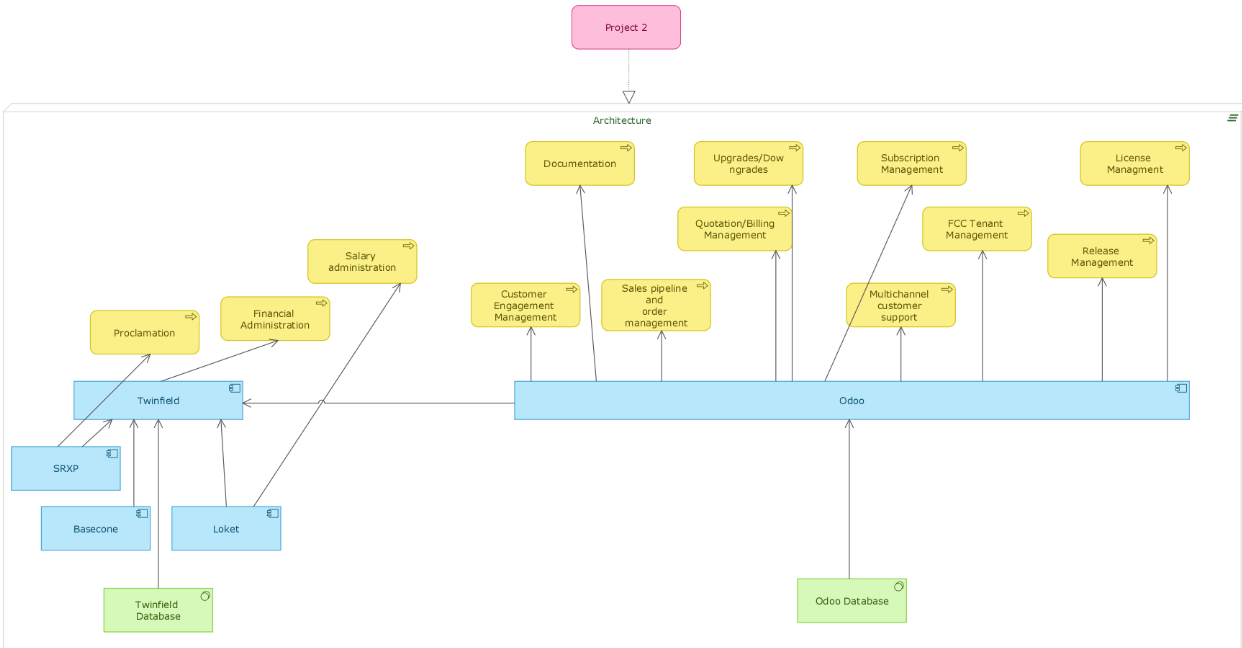


Figure 23. To-be Architecture Project 2

Project 3

The third project also preserves Twinfield for the financial administration but also preserves Wordpress. Wordpress is used for documentation, but also in the customer engagement process. With Hubspot phased out, a new connection should be made to Odoo to facilitate the link between the CRM and the website. Without phasing out Wordpress, the estimated project implementation time is 1,25 years.

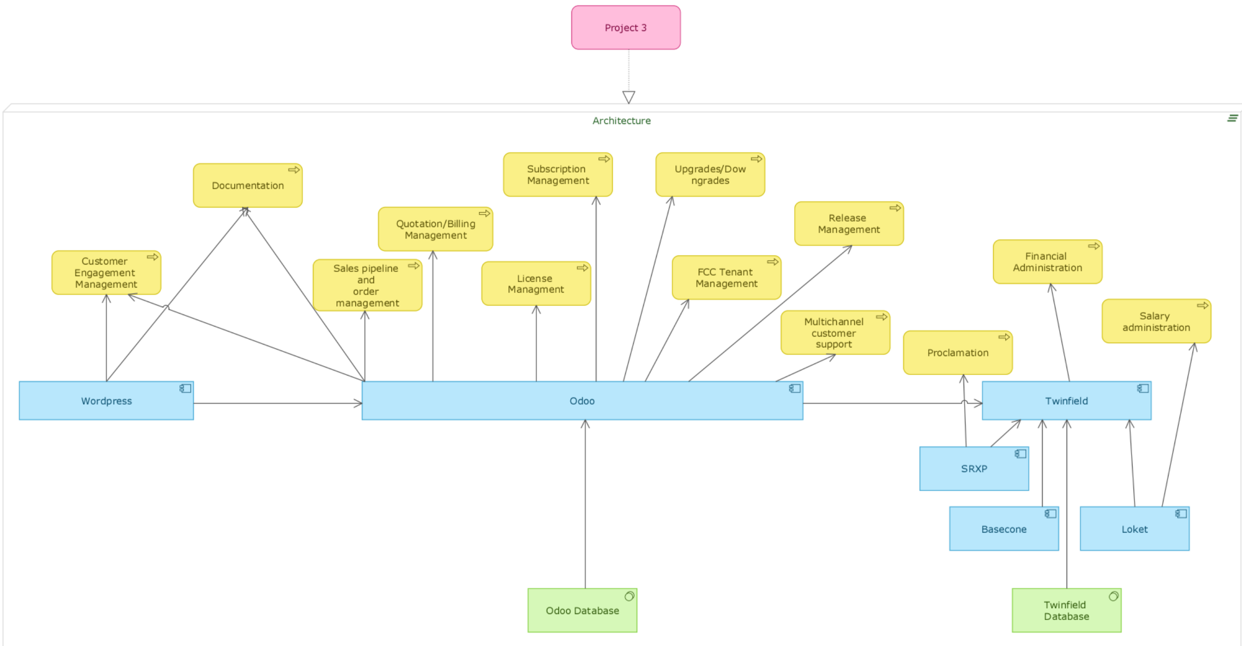


Figure 24. To-be Architecture Project 3

Project 4

The fourth project also preserves Zendesk. Zendesk is used as a customer portal where also support tickets can be administrated. Through this channel, upgrades and downgrades are often requested and are also used to communicate licenses to on-premise customers. Without phasing out Zendesk, the estimated project implementation time is 1 year.

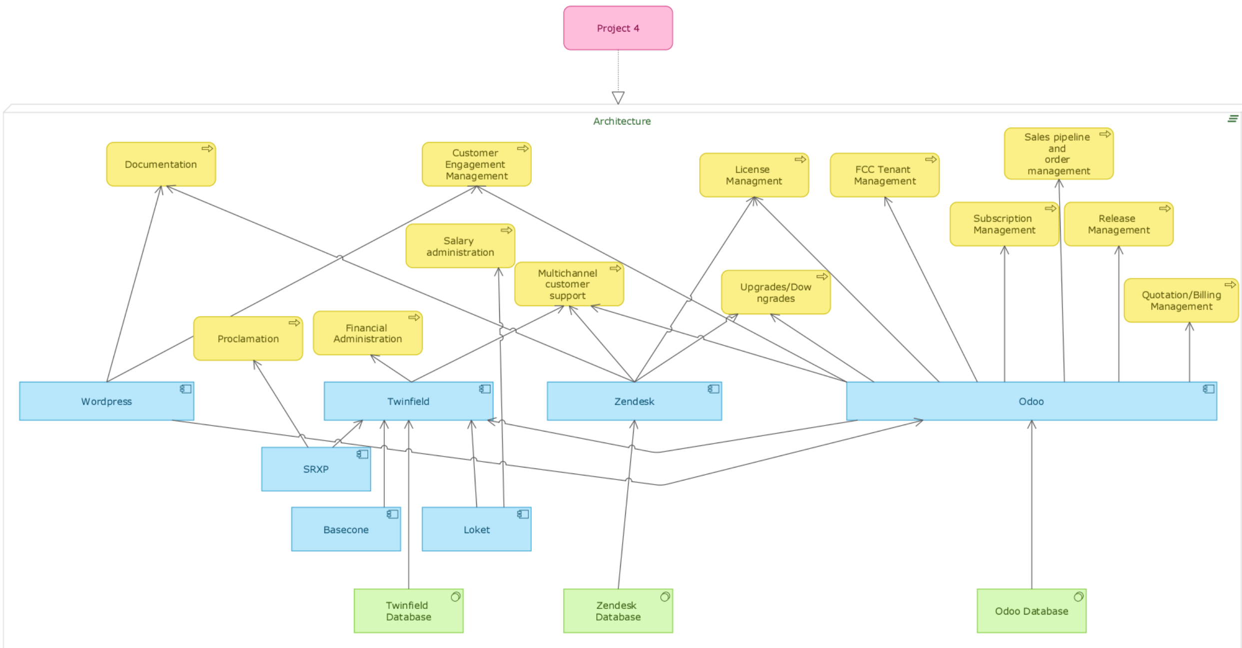


Figure 25. To-be Architecture Project 4

Project 5

The fifth and last project is the most preservative project. The project proposes to only phase out Hubspot and Salesforce while leaving the rest of the applications in place. Several integrations need to be made again, and it does not have a big effect on the overall amount of applications. The estimated project implementation time is 4 months.

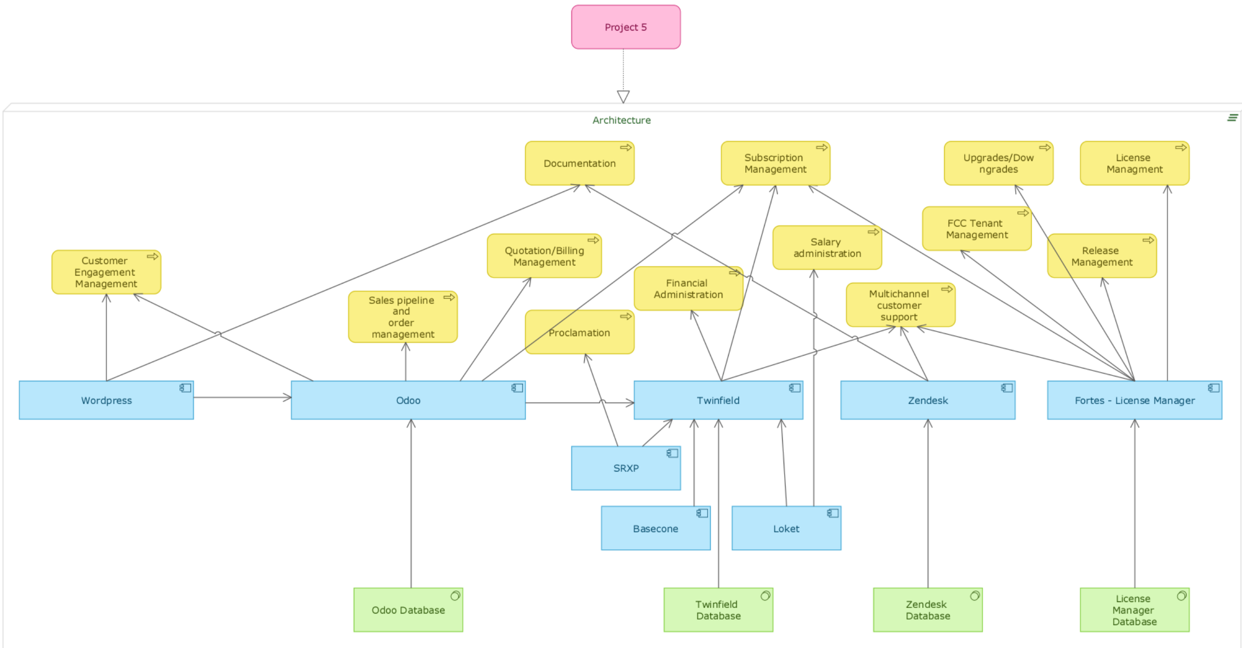


Figure 26. To-be Architecture Project 5

4.2.2 Enterprise Architecture Complexity Analysis

This section describes the application of the Enterprise Architecture Complexity Analysis on the case study of Fortes. The first step, determining the weights of the Enterprise Architecture Complexity Metrics, is described in section 4.2.2.1. The next step is calculating the Enterprise Architecture Complexity Metrics. The to-be architectures from the prerequisites are used. The third and last step in the Complexity Analysis is to calculate the impact of the project on the Enterprise Architecture.

4.2.2.1 Step 1: EACM Weight Analysis

As described in Chapter 3, the EACM Weight Analysis contains five steps. The next section will describe how this analysis can be applied in the context of this case study.

Step 1: Determine a set of decision criteria

The goal of this step is to determine the set of decision criteria that should be used. The operationalization of the previous section will be used as decision criteria. So, we have: { number_of_elements(c1), number_of_relations(c2), element_entropy(c3), relation_entropy(c4), element_modularity(c5), cost_of_rework(c6)}.

Step 2: Determine the best and the worst criteria

The goal of this step is to determine the best (most desirable, most important) and the worst (least desirable, least important) criteria. The criteria have different meanings. For Fortes, the goal is to minimize the number of systems/elements in the architecture and decrease the number of integrations between systems. Therefore, the number_of_elements(c1) and number_of_relations(c2) are perceived as most important criteria. On the other hand, the type of element is not relevant in this context, therefore the heterogeneity metrics are perceived as the least important criteria, so element_entropy(c3) and relation_entropy(c4).

Step 3: Determine the preference of the best over all other criteria

The goal of this step is to determine the weights of the best criteria over all other criteria with a number between 1-9, where 1 is equal, and 9 is the high preference of the best over the other. In our example, we preferred number_of_elements(c1) and number_of_relations(c2) equally. We can choose one of the two arbitrarily as best. The result is called the Best-to-Others matrix and can be found in Table 18.

	C1	C2	C3	C4	C5	C6
number_of_elements	1	1	5	5	3	2

Table 18. Best-to-Others matrix

Step 4: Determine the preference of all criteria over the worst

The goal of this step is to determine the weights of all criteria over the worst criteria with a number between 1-9, where 1 is equal and 9 is a high preference over the worst. In our example, we preferred element_entropy(c3) and relation_entropy(c4) equally. We can choose one of the two arbitrarily as worst. The result is called the Others-to-Worst matrix and can be found in Table 19.

	Element_entropy
C1	5
C2	5
C3	1
C4	1
C5	2
C6	3

Table 19. Others-to-Worst matrix

Step 5: Determine the optimal weights

The goal of this step is to determine the optimal weights of the criteria. A Microsoft Excel-based solver program is used. This solver program can be provided when this is requested. The solver gave the following result based on the input from the examples in the previous steps. The result can be found in Table 20. The weights are used in the Enterprise Architecture Complexity Analysis, which is the third step in this demonstration.

Criteria	**Weights**
number_of_elements	0,306
number_of_relations	0,306
element_entropy	0,058
relation_entropy	0,064
element_modularity	0,107
cost_of_rework	0,160
Ksi*	0,01455

Table 20. BWM Solver result

4.2.2.2 Step 2: EAC Score Analysis

This section describes the second step of the proposed model in the context of the previously introduced case study of Fortes. The goal of this step is to use the operationalization from Section 3.2.2.1 and use the metrics to measure the complexity of the as-is architecture and the available project alternatives. The result of this step is presented in Table 21. The calculation of the metrics is performed in detail for the first project.

Metric	**As-is**	**Score P1**	**Score P2**	**Score P3**	**Score P4**	**Score P5**
number_of_elements	27	15	20	21	23	25
number_of_relations	32	14	19	22	27	28
element_entropy	1,030409	0,485094	0,856841	0,878751	0,950209	0,997874
relation_entropy	0	0	0	0	0	0
element_modularity	0,4697	0	0,4252	0,4039	0,4001	0,5057
cost_of_rework	0	0	0	0	0	0

Table 21. Result of Enterprise Architecture Complexity Score Analysis

Number of elements

The calculation for the metric number_of_elements is given in chapter 3 as the total count of nodes in the architecture. The calculation of this step is therefore straightforward summing up all the different elements in the architecture. In the case of project 1: we have one element in the technology layer, one element in the application layer, and thirteen elements in the business layer, summing up to fifteen elements for Project 1, as stated in Table 21.

Number of relations

The calculation for the metric number_of_relations is given in chapter 3 as the total count of edges in the architecture. The calculation of this step is therefore straightforward summing up all the different relations in the architecture. In the case of project 1: there is one relation between the technological layer and the application layer, and there are thirteen relations between the application layer and the business layer. This sums up to a total of fourteen relations for Project 1, as stated in Table 21.

Element Entropy

The calculation for the metric element_entropy is given in chapter 3 as $-\sum_{i=1}^{n} p_i \ln(p_i)$ where p_i is the relative frequency of element type i. The type of element is needed in this case, including the relative frequency of the element. For project 1, there are thirteen business process elements in the architecture, one application component element and one system software element. The calculation states that for all types of elements, the relative frequency should be multiplied by the natural logarithm of the relative frequency. The sum should then be multiplied by -1 to get a positive score. In the case of project one, this results in -1 * (-0.1240207 + -0,1805367 + -0,1805367) = -0,485094 as stated in Table 22.

Type of element	Frequency	$p_i \ln(p_i)$
Business Process	13/15	-0.1240207
Application Component	1/15	-0,1805367
System Software	1/15	-0,1805367

Table 22. Element entropy calculation

Relation Entropy

The calculation for the metric relation_entropy is given in chapter 3 as $-\sum_{i=1}^{n} p_i \ln(p_i)$ where p_i is the relative frequency of relation type i. The type of relationship is needed in this case, including the relative frequency of the element. Project 1 is quite simple. The architecture consists of only serving relations. In this case, the entropy is 0. When this is not the case, a breakdown should be made per relation type, similar to how the element_entropy is calculated in the previous section.

Element Modularity

The calculation for the metric element_modularity is given in chapter 3 as $\frac{1}{4m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m}) s_i s_j$. The calculation of this metric is quite complex and requires mathematical knowledge that is outside of the scope of this research. We will not go into a lot of detail to calculate this particular score. For the sake of this case study, we choose to use R to calculate the modularity score by translating the architecture into a network where the elements are the nodes, and the relations are the edges. A code sample to do this can be found in Chapter 8. The modularity of the to-be architecture of project 1 is calculated with R and resulted in 0, the highest possible modularity, as the architecture is indivisible, also according to the theory of Newman (2006).

Cost of rework

The calculation for the metric cost_of_rework is given in chapter 3 as $\sum_k Cr(E_k)$. In principal, this means that the cost of rework is the propagated cost of rework over all elements in the architecture. Getting this insight is doable in an organization that has an accurate and up-to-date enterprise architecture. However, this is not the case at Fortes. Therefore it is quite difficult to calculate the cost of rework accurately in this case study. Therefore we will use 0 for all projects in this case study.

4.2.2.3 Step 3: Enterprise Architecture Complexity Analysis

The previous sections described the operationalization of Enterprise Architecture Complexity, how the Best-Worst Method can be applied to determine the weights of the metrics, and the last section determined the scores for all alternative projects. This section will describe how the results from the previous sections conclude in a single complexity number.

As described in chapter 3, we use the dimensional weighting method to determine the score. The calculated scores in step 2 are normalized and then multiplied by the weights from step 1 for each project alternative, including the as-is architecture scores. An alternative's total score is calculated by summing their ranks multiplied by the weight of the respective criteria. Then, the alternatives are ranked based on their total scores. The last step is subtracting the as-is situation complexity score from the to-be complexity scores to calculate the contribution of the individual projects to the overall architecture.

Metric	As-is	To-be P1	To-be P2	To-be P3	To-be P4	To-be P5	Weights
number_of_elements	1	0,556	0,741	0,778	0,852	0,926	**0,306**
number_of_relations	1	0,438	0,594	0,688	0,844	0,875	**0,306**
element_entropy	1	0,471	0,832	0,853	0,922	0,968	**0,058**
relation_entropy	0	0	0	0	0	0	**0,064**
element_modularity	0,470	0	0,425	0,404	0,400	0,506	**0,107**
cost_of_rework	0	0	0	0	0	0	**0,160**
Weighted Total	**0,719**	**0,331**	**0,502**	**0,540**	**0,614**	**0,661**	
Complexity Score		**0,389**	**0,218**	**0,179**	**0,105**	**0,059**	

Table 23. Normalized Results of Complexity Analysis and Complexity Scores

The output of this stage, the complexity score for each project, serves as input for the Project Selection Analysis in Section 3.2.4. The complexity score indicates the contribution of the project to the de-complexification of the architecture. Therefore, the score should be maximized.

4.2.3 General Project Analysis

The second stage of the proposed Project Portfolio Selection Process describes the General Project Analysis. In this stage, three common types of analysis are applied for project selection. Cost, risk, and benefit are analyzed in Section 4.2.3.1, Section 4.2.3.2, Section 4.2.3.3, respectively.

The outputs of the Prerequisites: the as-is architecture, to-be architectures, and the project information documents are used as input for the General Project Analysis stage. Cost, risk, and benefit analyses are applied and result in the following outputs: Cost Analysis Results, Risk Analysis Results, and Benefit Analysis Results. These outputs are used as input for the Project Selection Analysis in Section 4.2.4.

4.2.3.1 Step 4: Cost Analysis

The goal of the cost analysis is to determine the expected cost of the project. This analysis can be based on the difference between the as-is and to-be architecture, or other cost calculation methods. For this example, we will use the outputs of the prerequisites, namely the differences between the as-is and the to-be architecture. Only one of the five projects will be handled in detail to illustrate the application of the cost analysis. The other four projects are analyzed in the same way but are not included. As described in chapter 3, the cost analysis will be based on the differences between the as-is architecture and the to-be architectures. Costs are related to specific change events of the architecture. Possible change events are for example deletion or addition. Not only elements can be changed, but also the relations between the applications can be changed. The changes in architecture can be visualized like the presented example of the architectures of project 1 in Figure 27.

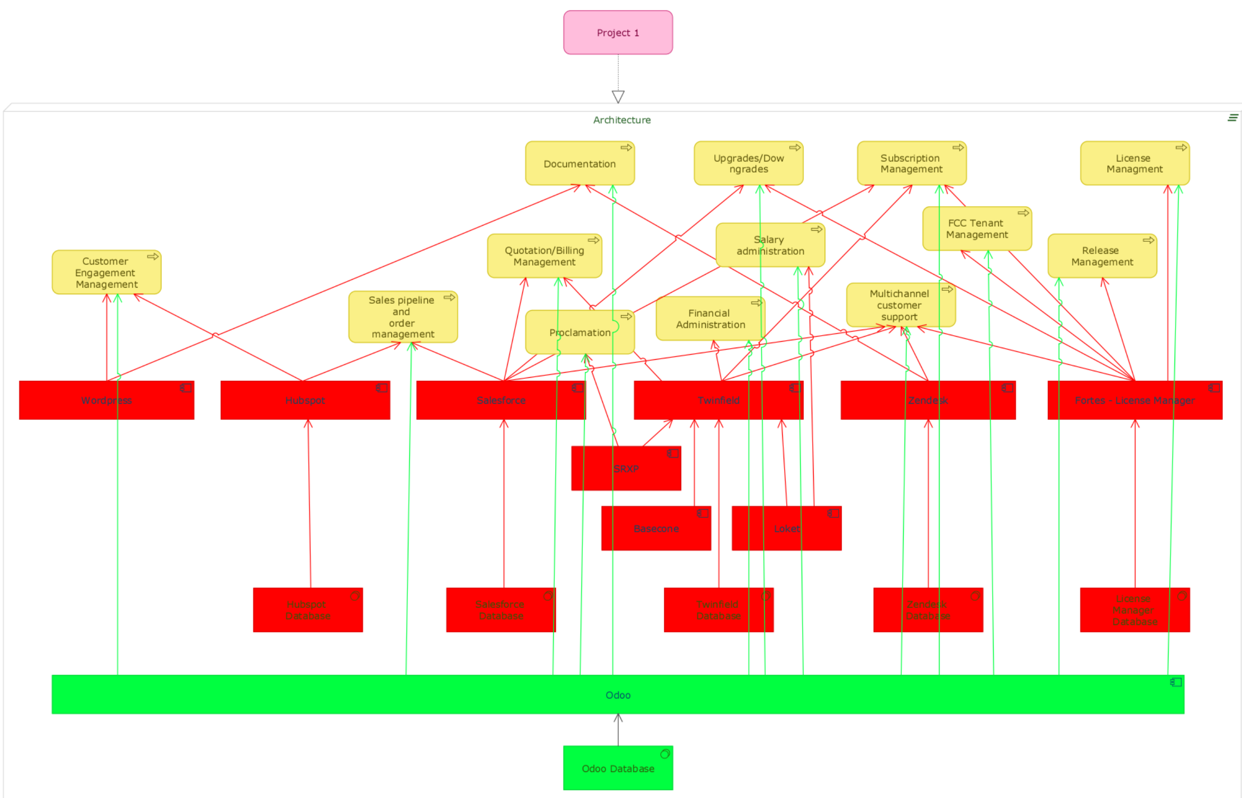


Figure 27. Impact Analysis

The architecture in Figure 27 clearly shows what types of change need to be executed. The red-colored elements and relations are deleted, the green elements and relations are added, and the yellow items are modified.

In chapter 3, the values of the changes are displayed in the architecture. For the case study, we decided to list the costs in a table per specific change event in the architecture. The results are presented in Table 24.

Description	**Time (Hours)**	**Costs (€)**
Deletion of Wordpress	10	
Deletion of Hubspot	10	
Deletion of Salesforce	10	
Deletion of Twinfield	10	
Deletion of SRXP	10	
Deletion of Basecone	10	
Deletion of Loket	10	
Deletion of Zendesk	10	
Deletion of License Manager	10	
Addition of Odoo	80	7300
Support from an experienced Odoo consultant	40	4000
Modification of Customer Engagement Management	320	
Modification of Salespipeline and Order Management	320	
Modification of Documentation	480	
Modification of Quotation and Billing Management	240	
Modification of Proclamation	80	
Modification of Financial Administration	480	
Modification of Salary Administration	120	
Modification of Upgrades/Downgrades	240	
Modification of Subscription Management	320	
Modification of Multichannel Customer Support	180	
Modification of FCC Tenant Management	240	
Modification of Release Management	80	
Modification of License Management	200	
	+- 3500 hours	**€7300**

Table 24. Cost Analysis Case Study Project 1

From the results of Table 24, we can conclude that most of the costs are measured in hours. Only the addition of Odoo has a specific monetary value attached to it, as it involves a software-as-a-service subscription and some external consultancy. There is the option to multiply the total hours with a price per hour and sum the result with the calculated monetary costs. In the case of Fortes, the project is performed internally. Therefore, we choose to only include the specific monetary value in the cost calculation. The required implementation hours are also reflected in the implementation time from the Project Information Documents. Therefore, the hours are excluded from the analysis. The total costs are in the case of Project 1 €11.300.

4.2.3.2 Step 5: Risk Analysis

The goal of the risk analysis is the identification of risks of the project. As described in chapter 3, the risks will be based on the change events in the architecture. These changes in architecture are also used in the Cost Analysis in Step 4. The Impact analysis in Figure 27 will be used again in this step.

An overview of the identified risks for Project 1 is presented in Table 25. The risks were presented at the case study company and scored on the impact on the probability of occurring. The risk factor is then calculated with the risk matrix as described in Chapter 3.

	Impact	**Probability**	**Risk Factor**
Integration failure with business processes	4	4	3 (high)
Data corruption/loss during migration	5	4	4 (critical)
Asynchronous data due to two systems running parallel	3	5	3 (high)
Errors in financial administration due to migration	6	6	4 (critical)
Angry clients due to migration of documentation channels	2	3	2 (medium)
Project incompletion	6	3	4 (critical)
Loss of website effectiveness due to migration	4	3	3 (high)
		Average	**3,3 (high)**

Table 25. Risk Analysis Case Study Project 1

First of all, the ERP system will influence all identified business processes. A lot of applications were involved in the processes and were customized over the years. Trying to map those processes to the ERP and changing the processes on the fly is a complicated task with the risk of integration failure. The impact is quite high and the probability as well. Those different applications also stored their data. All that data should be migrated to the ERP system in the right format. This could lead to data loss and corruption. The impact is very high and the probability is high as well. During the migration, a hybrid situation exists. The migration is not done within a day, and operations proceed. Therefore, data should be synchronized or administrated twice during that period. This leads to the risk of having asynchronous data due to the two systems running parallel. The impact of this risk is medium, while the probability is very high. A complex application in the as-is architecture is Twinfield, which is used for financial administration. This application is highly integrated with other applications and is used by external parties for example bookkeeping and accountants, while it also reports the operational results to the shareholders of Fortes. In other words, there is a high risk of migrating this application due to all its dependencies and interwovenness with other processes. Therefore, the risk of errors in the migration of the financial administration is classified with the highest score for impact and probability. The information sources for clients will also be migrated. Clients that do not like that change will not be happy with that. The risk is classified with low impact and medium probability. With a project of this size, the probability of not completing the project due to several possible circumstances is not negligible. The impact of not completing the project is critical, as not finishing the project will leave the organization in a volatile state. Also, not completely implementing an ERP system will not result in the projected results and will not unleash the possible potential of such an integrated system. The probability of this risk is medium, as the organization is highly invested in completing the project, and will do anything to finish it. The migration of the website can result in an effectiveness loss, as it has build up a reputation in the last years and works as an effective marketing machine. Rebuilding the website can result in a loss of effectiveness. The impact of that is high as it would result in less qualified leads, but the probability is estimated to be medium.

4.2.3.3 Step 6: Benefit Analysis

The goal of the benefit analysis is to determine the expected benefits of the project. This analysis can be based on the difference between the as-is and to-be architecture, or other benefit calculation methods. For this example, we will use the outputs of the prerequisites, namely the differences between the as-is and the to-be architecture. Only one of the five projects will be handled in detail to illustrate the application of the benefit analysis. The other four projects are analyzed in the same way but are not included.

As described in chapter 3, the benefit analysis will be based on the differences between the as-is architecture and the to-be architectures. Benefits are often calculated as concrete monetary values. However, this is often not applicable, as benefits are mostly focussed on the long term. Little differences can have large effects in the future. Therefore we use the proposed classification as specified in chapter 3. All benefits are classified with a number between one and five, where five is the most important. Also, this step will be applied to one project alternative of the five possible projects in the case study. The benefits of project 1 are analyzed.

The environment in which Fortes operates, the project portfolio management software environment, is changing rapidly. It requires that these companies can give their customers integrated experiences between different disciplines. For example, the customer wants to be able to see its current SaaS subscriptions, upgrade the subscription, or request an upgrade. But also support should be integrated there. Currently, Fortes uses a different business application for each discipline. This makes it impossible to create an integrated experience for their customers. Also, it is hard to integrate processes and automate them. For example, when a customer signs a contract, automatically a SaaS product should be brought online and it should be communicated to the appropriate person. Trying to integrate these kinds of processes between different applications is a very difficult job, and takes a lot of maintenance time. Putting all disciplines in one application that supports the integration of certain processes creates the possibility for the organization to move much faster and work more efficiently, leaving more time to innovate. Recorded data is in this way also aggregated in one application. The fragmented data sources can then be replaced with a central database that stores all data. Creating reports across disciplines, for example checking whether clients with a lot of active users in the SaaS application of Fortes also report more support tickets, is now a matter of seconds. With the old situation, several reports should be exported in Excel, and then in some way combined and analyzed. Getting those insights live would be near to impossible. Replacing the old applications with one system seems like a very beneficial project. A graphical representation as suggested in chapter 3 is represented in Figure 28.

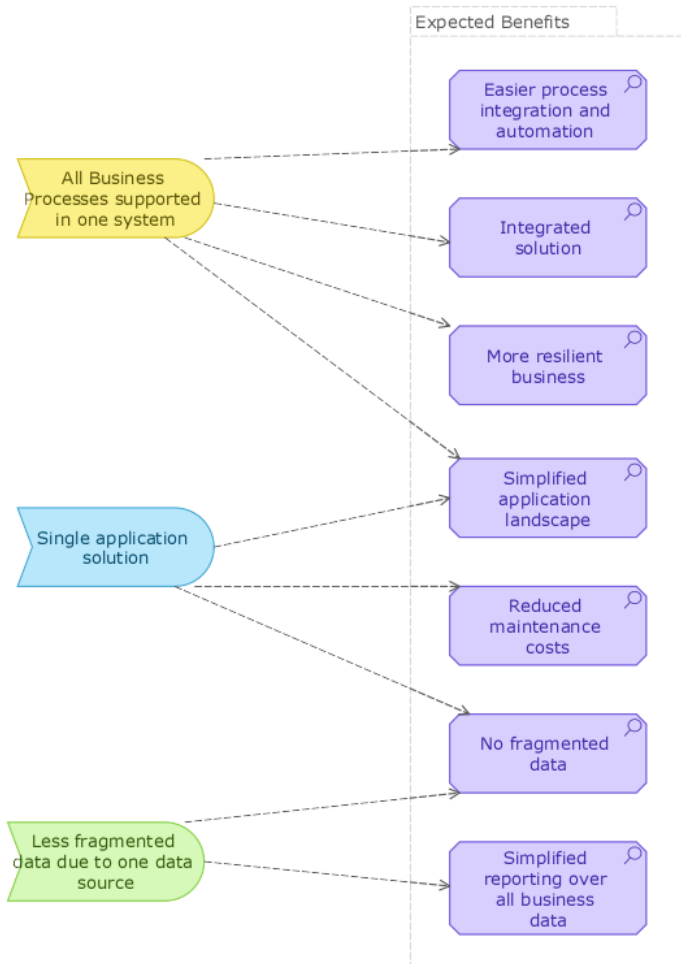


Figure 28. Benefit Analysis Case Study Project 1

The identified benefits from the first project are identified and presented in Table 26. The scores are averaged, resulting in the benefits score for the first project. This type of analysis is also performed on the other projects, but not included.

Benefit	**Score**
Easier process integration and automation	5
Integrated solution	4
More resilient business	5
Simplified application landscape	4
Reduced maintenance cost	4
No fragmented data	4
Simplified reporting on business data	5
Average	**4.4**

Table 26. Benefits Analysis Case Study Project 1

4.2.4 Step 7: Project Selection Analysis

The first step of the Project Selection Analysis is combining the output of all the previous steps for all the project alternatives. Table 27 represents the decision criteria of the five projects, as presented in the previous sections.

Criteria	P1	P2	P3	P4	P5
Complexity Score	0,388563	0,217883	0,178959	0,104958	0,058819
Cost	11300	10900	10300	10100	10100
Risk Score	3,3	3	2,6	2,3	2
Benefit Score	4,4	4,2	3,8	3,2	3
Project Duration	24	18	15	12	6

Table 27. Project Selection Criteria Scores

For the five criteria, the scores of the four project alternatives are compared. Then, after applying the Best-Worst Method five times on the different criteria, the weights of the criteria are summed up per project alternative, as described in the dimensional weighting method by Russel et al. (1987). The highest summed-up score is the best (most suitable) project.

To illustrate the application of the Best-Worst Method, we will apply it to one of the criteria. The five steps of the Best-Worst Method are applied to the Complexity Score. The method should be applied for each one of the five criteria.

Step 1: Determine a set of decision criteria

The first step of the Best-Worst Method is determining the set of decision criteria. As described, the decision criteria, or the project alternatives, are {P1, P2, P3, P4, P5}.

Step 2: Determine the best and the worst criteria

The goal of this step is to determine the best (most desirable, most important) and the worst (least desirable, least important) criteria. The best and worst are in this case dependent on what criteria are under consideration. The Complexity Score should be maximized, as it indicates the contribution to the de-complexification of the architecture. The best criteria/project is in this case the project with the highest Complexity Score, which is P3. The worst alternative is in this case P5 for the Complexity Score. In the case of costs, the goal is to minimize. In that case, the smallest cost should be considered as the best alternative.

Step 3: Determine the preference of the best overall criteria

The goal of this step is to determine the weights of the best criteria over all other criteria with a number between 1-9, where 1 is equal, and 9 is the high preference of the best over the other.

	P1	P2	P3	P4	P5
P1	1	5	6	7	8

Table 28. Best-to-Others matrix

Step 4: Determine the preference of all criteria over the worst

The goal of this step is to determine the weights of all criteria over the worst criteria with a number between 1-9, where 1 is equal and 9 is a high preference over the worst. In this example, P4 is considered the worst. The result is called the Others-to-Worst matrix and can be found in Table 29.

	P5
P1	8
P2	4
P3	3
P4	2
P5	1

Table 29. Others-to-Worst matrix

Step 5: Determine the optimal weights

The goal of this step is to determine the optimal weights of the criteria. Again, the Microsoft Excel-based solver program is used. This solver program can be provided when this is requested. The solver gave the following result based on the input from the examples in the previous steps. The result can be found in Table 30. The table also shows the weights of the other criteria.

Criteria	P1 Weight	P2 Weight	P3 Weight	P4 Weight	P5 Weight	Sum
Complexity Score	0,587	0,138	0,115	0,099	0,060	1
Cost	0,071	0,118	0,176	0,318	0,318	1
Risk Score	0,112	0,131	0,196	0,196	0,364	1
Benefit Score	0,462	0,270	0,135	0,090	0,043	1
Project Duration	0,072	0,118	0,158	0,237	0,416	1
Total	**1,304**	**0,775**	**0,781**	**0,939**	**1,201**	

Table 30. Weights and Sum per Project

The goal of this step is to get a prioritized list of alternatives based on the criteria. The prioritization is based on the sum of the weights of the different criteria per project. In this case, the best alternative project is P1 with a total score of 1,30392. The prioritization of the projects can be found in Table 31.

Project	Score
P1	**1,304**
P5	**1,201**
P4	**0,939**
P3	**0,781**
P2	**0,775**

Table 31. Prioritized projects

5 EVALUATION

This chapter describes the evaluation of the proposed method in Chapter 3. The evaluation phase is part of the Design Science Research Methodology and is intended as an evaluation step to measure the quality of the proposed method towards supporting specific objectives. The evaluation is done by performing expert interviews with a selection of people from Fortes and their partner network. The proposed method was introduced. The interview is conducted after the introduction of the method. First, we introduce the conceptual basis of the interview, followed by the interview questions and the sample selection, concluding with the results of the evaluation.

5.1 UTAUT

In Design Science, the goal of the Evaluation step is among other things to find out if the method could be generally accepted by users for potential use in practice. Some subject matter experts are interviewed to evaluate the potential use of the method in practice. To formulate the questions for the interview, the concept of Unified Theory of Acceptance and Use of Technology (Venkatesh et al., 2003) is adopted. Although the proposed method is not recognized as a technology, or more specifically a software program, the constructs of UTAUT are relevant for the evaluation of the method. UTAUT explains the distinct aspects of the behavior of the users and the acceptance of it. The concept UTAUT considers multiple constructs that are proven to have a significant effect in the determination of user acceptance and usage behavior, such as performance expectancy, effort expectancy, and experience. These constructs are also interesting in the evaluation of the method. Therefore, the UTAUT is used to evaluate the method. The UTAUT model can be found in Figure 29.

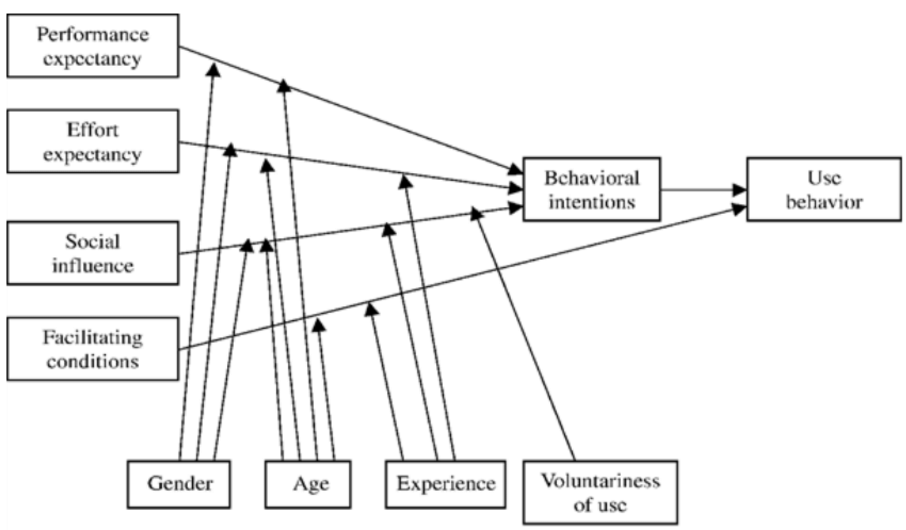


Figure 29. UTUAT (Venkatesh et al., 2003)

The UTAUT concept recognizes performance expectancy, effort expectancy, social influence, and facilitating conditions as constructs with a significant role. Also, additional constructs play a role in determining the UTAUT, namely gender, age, experience, and voluntariness of use. These aspects have a moderating role towards user acceptance and are indirect determinants of intention.

Table 32 represents the UTAUT constructs that are considered relevant for this research and will be used in this research to validate the proposed method. The questions corresponding to the constructs as defined by Venkatesh et al. (2003) are adapted to the use in this research at the discretion of the researcher.

Construct	Definition
Experience	The level of experience of the individual
Performance expectancy	The degree to which an individual believes that using the system will help him or her to attain gains in job performance
Effort expectancy	The degree of ease associated with the use of the system
Attitude towards using Technology	An individual's overall affective reaction to using a system
Facilitating conditions	The degree to which an individual believes that an organizational and technical infrastructure exists to support the use of the system
Behavioral intention to use	A person's perceived likelihood or the subjective probability that he or she will engage in a given behavior

Table 32. UTAUT constructs (Venkatesh et al., 2003)

5.2 Interview

Each interview was approached in the same way. The interview was initiated with a short overview of the method. The different stages are explained, and the input and output of each step are illustrated. When the overall method was clear and understood, the case study was presented. The full method is applied. Starting with the problem and the definition of the project alternatives. The current Enterprise Architecture is presented to the interviewees, followed by the to-be architectures of the identified projects.

The complexity analysis was presented completely. Starting with a general explanation of MCDA. Then, the weight analysis was explained. The different architectures are scored on the complexity metrics in the score analysis. The outputs of the previous steps are combined in the complexity analysis, resulting in a complexity score per project.

The general project analysis is not explained in a lot of detail, as the interviewees already have lots of experience with those types of analysis. The idea of determining costs, risks, and benefits based on the enterprise architecture is presented and explained.

The final step, the project selection analysis, starts with the different inputs. Then, the MCDA is explained again, but now in the context of scoring the different projects on their contribution to the criteria. The MCDA is applied for each criterion. Then, the weights are applied and the list of prioritized projects is presented. The interviewee is asked whether it is in line with the expected outcome of the prioritization.

Possible questions are answered before starting the evaluation questions. Till this point, the interview is estimated to take around twenty minutes, leaving forty minutes for the evaluation questions. The evaluation questions are based on the UTAUT.

UTAUT is usually a method to quantitatively analyze the use and adoption of technology. The goal of this step is to qualitatively analyze the method. This requires some adaption of the defined questions. The questions from UTAUT are defined in Table 33. The questions form a basis for the interview. Follow up questions can be asked if it is deemed necessary by the researcher. The interview is therefore semi-structured. The duration of the interview is around one hour.

Construct	Questions
Experience	What is your experience with Project Portfolio Management? What is your experience with Enterprise Architecture?
Performance Expectancy	Do you think the measurement model for Enterprise Architecture Complexity is accurate? Do you think the inclusion of Enterprise Architecture Complexity as a criterion improves the Project Portfolio Selection? How?
Effort Expectancy	Would your interaction with the method be clear and understandable? Would it be easy for you to become skillful at using the method? Would you find the method easy to use? Would learning to operate the method be easy for you?
Attitude towards using Technology	Is using the method a bad/good idea? Do you think the method makes work more interesting? Do you think working with the method is fun? Would you like to work with the method?
Facilitating conditions	Do you have the resources necessary to use the method? Do you have the knowledge necessary to use the method? Is the system compatible with other methods you use?
Behavioral intention to use	Are you intending to use the method in the next 12 months? Are you predicting to use the method in the next 12 months? Are you planning to use the method in the next 12 months?

Table 33. Interview Questions

5.3 Evaluation Analysis

The interviews were conducted via Microsoft Teams. Although the research is in English, the interviews are conducted in Dutch. The interviewees are all Dutch and are less able to express themselves in English. Therefore the researcher assessed that the interviews should be conducted in Dutch such that the results will not be of less quality due to the language barrier. The quotes are translated into English. The introduction of the method is not recorded. The recording is started at the start of the evaluation questions. The interview is recorded because it helps the interviewer keep his focus on the interviewee and the questions. The recordings are transcribed afterward. The transcriptions are analyzed by using the general inductive approach, as presented by Thomas (2006).

5.4 Sample Selection

The sample of interviewees is selected from the employees of Fortes. However, not all employees have the necessary experience with project portfolio management to evaluate the model. Therefore, only employees have selected that know about Project Portfolio Management. The sample includes consultants and product managers. The interview results are anonymized. Table 34 represents the role of the interviewee and the duration of the conducted interview.

Interview	Role	Duration
1	Consultant	00:49
2	Consultant	01:18
3	Consultant	01:08
4	Product Manager	01:04
5	Product Manager	00:50

Table 34. Evaluation Sample Roles and Duration

5.5 EVALUATION RESULTS

This section describes the results of the interviews that are conducted as part of the evaluation of the project portfolio selection method as proposed in Chapter 3. The following subsections follow the reporting style of the results that are the result of the general inductive approach by Thomas (2006). This reporting style constitutes a label for the category, the author's description of the meaning of the category, and quotations from the raw text to elaborate the meaning of the category and to show the type of text coded into the category.

Categories	**Prevalence**
Use in practice	100%
Objectiveness	100%
Usability	100%
Automating	100%
Validity	80%
Knowledge/experience	80%
New insight	60%
Resilience	20%
Novelty	20%
Optimization	20%

Table 35. Evaluation Categories with Prevalence

The following sections describe the most prevalent categories induced from the interview transcripts. Section 5.5.8 shortly presents the less prevalent categories. The threshold was set at more than twenty percent. Things that are brought up only in one interview are considered not prevalent enough to be included extensively and are therefore only described shortly.

5.5.1 Use in practice

The category "use in practice" describes if and how the method can be used in practice.

Talking about the ability of organizations to work with the method, *"low maturity organizations, which are a lot of organizations, will most likely not be able to use the method"*. This is an expectation and raises the question *"to what extent is organizational maturity necessary to apply this method?".* A problem could be that *"98% of the companies would not have these architectures on the shelf and would have to produce them on the fly".* This is quite a problem, as having these architectures is a prerequisite for the model, and *"the method is only helpful when the company has an up-to-date enterprise architecture".* Not having these architectures is one thing, but *"most organizations would be unable to deliver such architectures"* and *"it's a theoretical approach that cannot be used in practice shortly. They want to but are unable to".* However, there is also light in the tunnel, as *"analyzing what changes and what gets replaced is part of a good information analysis"* and *"the method can be used in practice if the information analysis is of good quality".*

The fact that not a lot of organizations are able to use the method, does not mean it is not useful or efficient to use. On the contrary, *"formalizing this method will reduce the workload of critical resources such as architects"* as *"the results are much faster to grasp than policy documents and other judgment documentation about the projects"* because *"comparing based on descriptions is a lot more work then* *analyzing the results from the method".* Therefore *"using the method would increase the capacity of the critical resources such as architects".*

A problem was raised during an interview. *"I have to think really hard to think how to apply the method on fundamentally different projects"* because *"the case study chose between six alternatives, not six fundamentally different projects".* The case study showed that *"it is a suitable method for alternative selection"*, but the case study *"focusses on what changes in each scenario",* and the method was *"used for analyzing alternatives of a project".* Therefore *"I think that you need alternatives within the same playing field to make the comparison".* Applying the method to alternatives is not always possible as *"alternatives are not extensively described in the business cases".* However, not all interviewees agree, and think that *"the method can be used on both alternative and project selection".*

Currently, *"organizations look at the architecture, but not as structured as proposed in the method".* However, *"a uniform approach to value projects is extremely important"* as *"more organizations also ask an enterprise architect's opinion"*. But *"business analysts use more specified methods than enterprise architects at this moment"*. A lot of organizations don't look quantitatively at projects as *"I advise my clients to think about how projects can be judged quantitatively"*, but that is not always easy because *"setting up a quantitative project scoring method in an organization is difficult".* That does not mean that those organizations do not want to use quantitative methods, because *"a lot of people would like to work with such a method, especially in complex organizations"*. *"The method would be most applicable in very complex organizations with hundreds of applications".*

The interviewees think it is a valid method because *"if I was a portfolio manager I would seriously consider using the method"*, but *"the value of the method is in the complexity analysis, as it will fit the methods that are currently used by organizations". "The method can be used with existing methods as it is common to judge based on multiple aspects. This method outputs a number, which can be used next to other quantitative analyses"*, because *"using weighted scores is used often and enables adding the complexity to existing methods".* Being able to use the complexity analysis improves possible adoption, as *"the included criteria in project selection differ per organization".* An important effect of using the method is that *"the method would enable setting architecture de-complexification on the agenda"*. An interviewee states that *"the model will have more value in complex organizations"* and that *"a not so dynamic environment would not benefit from using this method".* However, *"due to digital transformation, this method is also increasingly interesting for smaller organizations".*

5.5.2 Objectiveness

This category describes how portfolio selection can benefit from the objectiveness of the complexity score. The method is perceived as *"an objective measurement instrument"* and a *"good method to objectively measure complexity". "It's a good objective method to start your decision process which is often not included in the information analysis". "It helps to look at projects objectively which is where a lot of money is lost when excluded".* But it also helps the persons trying to sell the project to decision-makers, as *"objectivity helps to prove the projects' contribution".* This is necessary because *"in practice, projects are often subjectively chosen"* and there is *"lots of subjectivity in existing project selection".* This is sometimes a problem, because *"subjective scoring is doable in small organizations but not when hundreds of systems are involved".* "*Using the method would minimize subjectivity in project selection",* which is a good thing, as *"excluding subjectivity is not a problem as project selection does not benefit from it".* One interviewee mentioned that *"calculating the cost of rework is difficult to measure objectively"* as it is not provided with formal parameters what to classify as a cost of rework. *"Subjectivity needs to be extracted from the cost of rework"* to make the method even more objective.

5.5.3 Usability

This category describes how usable the method is perceived by the interviewees. The perceptions are split. *"The method does not sound very complex". "The method is logical and does not need hours to look through"*. One interviewee nuanced it by saying that *"Architects will grasp the model easily".* But not all interviewees agreed. *"I cannot reproduce the model now which illustrates its complexity". "The method is usable for me, but likely not for many others". "The model is not easy to comprehend, I had to ask a few questions and look closely to follow". "Filling in and applying the method and determining the weights are difficult to do well".* However, it is also dependent on the type of organization, because *"in less exact organizations the method will be perceived as difficult"*. The use of the method is also dependent on the documentation. *"The method needs thorough documentation to be used".* It would also help the adoption in practice. *"Good documentation would help adoption in practice".* But it is probably not enough. One interviewee states that *"organizations would require additional help to use the method".* Organizations often don't have those architectures available, which would make it difficult to use the method, as it would require the creation of architectures in the business cases of projects.

5.5.4 Automating

This category describes whether the method should and could be automated in some way. One interviewee stated that *"calculating the metrics by hand is complex and not doable for me".* Currently, most of the calculations are done by hand. The interviewees unanimously think that *"automating the method is doable"* and that *"automating the method would make the method easier to understand".* But not only understand but *"adding automatic calculations would make the method easier to execute for everyone"* as well. *"Automating the calculations would help using the method in practice"* as *"automated calculations would make applying the method be just a filling-in-problem".*

5.5.5 Validity

This category describes whether the method is considered to be valid. In general, the method is received positively. Some interviewees mentioned that *"the method is clean and worthwhile"* and that it *"covers the right aspects of complexity".* The criteria in the complexity analysis are perceived as properly chosen. *"The criteria give a proper indication of what complexity is".* There is a balance between the effort that is necessary to use the method, and the accuracy of the output, as *"a larger accuracy would result in much more work".* Some interviewees stressed that *"the validity of the method is dependant on the number of involved people",* because *"if only one person is involved in the application of the method, it introduces possible favor for certain solutions".* The application of the method in the case study was perceived positively, but *"the method requires further evaluation and finetuning in practice".* One interviewee was wondering if the measurement model was correct and *"whether more elements result in linearly more complexity",* as the model currently just looks at the number of elements in the architecture. Another interviewee stated that the calculation of the cost of rework is possibly not objective enough if no formal specifications are given what to classify as cost of rework. This could open possibilities to favor a preferred project by inflating the cost of rework of other projects. In the project selection analysis, all criteria are weighted equally, but *"maybe the project selection analysis also requires weighting on its criteria, as complexity may be more important than costs".*

5.5.6 Knowledge/experience

This category describes what knowledge or experience is required to comprehend or use the method. The method is perceived inconclusively. While most interviewees perceived the method as complex, one did not think the method was complex. "*No steep learning curve is required to comprehend the method".* However, they did agree on the need for experience to be able to create the architectures. *"Extensive knowledge of the context is required to create the architectures". "Experience is needed to create the to-be and as-is architectures".* The issue is also raised that the method is only understandable for architects, or at least, it becomes less understandable when you get further away from the architects. "*The further from the architects, the harder the method becomes".* An interviewee also stated that experience with the method is required to apply the method correctly. "*You would need some experience to use the method the right way".* Another interviewee saw a connection between the experience of the user and the accuracy of the method. "*The knowledge and experience of the user of the method is a big factor in the success and the accuracy of the method".* Also, the results of the method were considered hard to grasp and would require knowledge about the context. *"Results are not conclusive and prioritization requires knowledge about the context".* The results were also considered hard to interpret by one interviewee. *"The results are not easily interpretable".*

5.5.7 New insight

This category describes the new insights that are discovered by applying the method. Some of these insights refer to the results of the case study that were mentioned before the interview. *"The case study had surprising results, I expected another prioritization". "The results of the case study are surprising and result in new insight".* The interviewees mentioned that *"the method requires to look at projects more in relation to the architecture"* and that *"using the method will start discussions within the organizations".* But besides that, *"it is an extra component to develop your opinion of projects and make your choices more robust".* It helps organizations because *"the method makes a more founded choice possible in the project selection for organizations that depend heavily on their IT infrastructure".*

5.5.8 Other

The last three categories describe the least prevalent subjects that were identified in the interview results. All three categories are mentioned only in one interview, and will not be described in detail.

Business resilience was mentioned in relation to enterprise architecture complexity. *"It is underestimated what the effect is of a complex architecture on the resilience of organizations".* The interviewee also proposes *"to add a factor that measures how more resilient the company became".* Resilience also tells us something about the ability to adapt to future change. *"Judging projects should not only be done on the current business situation but also to the business ability in the future and its adaptability to change".* The interviewee identified a relationship between complexity and resilience. *"Less complex architecture results in a more resilient business".*

The novelty of the method is also mentioned. This type of analysis is not used often in practice, as *"current judgment in practice on architecture is not as quantitative, nicely visualized and based on numbers".* It is taken even further, *"the approach as suggested in this method is not used in practice".* But that does not mean that those companies don't want to change because *"all organizations want something like this but they don't have it".*

Possible optimization of the parameters in the method is also mentioned. *"Optimizing the method would improve the way it is used"*. A possible approach for this optimization is also suggested. *"The method could be optimized in cycles like in a PDCA cycle. Then you should look at the realized benefits and feed that back to the way the method is used".* This is an interesting angle for organizations *"to make sure the scores become more accurate".*

6 CONCLUSION

This chapter describes several aspects of the research. First, Section 6.1 describes in short the thesis and concludes each chapter that is presented in this thesis. Section 6.2 presents the contributions of this research to theory and practice, followed by its limitations in Section 6.3 and future work in Section 6.4.

6.1 DISCUSSION

The main objective of this research is to design a project selection method that incorporates enterprise architecture complexity. Consequently, this objective is depicted in the scope of this research which is derived from the formulated research question: *How to design a Project Portfolio Selection Method that uses Enterprise Architecture Complexity as a decision criteria?* The proposed method consists of seven steps that are discussed in detail. Several methods are introduced that can be used in the various analyses. Those methods include a novel approach towards calculating enterprise architecture complexity by means of the multi-criteria decision analysis method called the Best-Worst Method, and enterprise architecture-based cost, risk, and benefit analysis.

The main research question is split up into three sub-questions, from which the first one is answered in the systematic literature review in Chapter 2. The first sub-question is formulated as follows: *What is the current state of the art of Project Portfolio Management and Enterprise Architecture, and what Project Portfolio Selection Method is most suitable for integration with Enterprise Architecture Complexity?* This question is quite broad and contains a few elements. Therefore the question is split up into four different questions in the literature review. The first two questions regard the state of the art of project portfolio selection methods and enterprise architecture complexity. The literature review clearly showed that both disciplines consist of several processes and are essential for organizations. Both functions strategically align to the business and control subordinate functions on the tactical level to maintain consistency for future changes. While EA concentrates on IT projects, PPM encompasses all major changes of the enterprise. Both sides analyze potential projects based on their needs. These analytical outputs need to be exchanged and discussed between EA and PPM to achieve a joint way forward. This research is a step in the direction of cooperation between EA and PPM. The third question focuses on what methods are most suitable to use when integrating enterprise architecture complexity in the project selection process. It became clear that a quantitative approach was most suitable in this case, and that MCDA methods are widely used. The most popular method is the AHP. However, some articles also mentioned its required effort to use and suggested other approaches, such as the BWM which is used in this research as it is a lot less work to use and it produces more consistent results. The fourth question was formulated to research how enterprise architecture complexity can be quantified and measured with the use of available metrics. Formal enterprise architecture complexity calculations were not found in literature, but several metrics are identified, and some research also indicated a more structured measurement model. Relevant parts of the measurement model are used to base the complexity calculation on. This chapter formed the basis for the proposed method and outlines the several design choices that had to be made.

The second sub-research question is answered in Chapter 3 and is formulated as a design problem which states: *How to design a Project Portfolio Selection Method with Enterprise Architecture Complexity Metrics?*. The proposed method requires some prerequisites, such as the as-is enterprise architecture, a defined problem with a list of projects, accompanied by the target architectures of the defined projects. The method heavily depends on the enterprise architectures and requires them to be explicitly modeled. The evaluation showed that this can be a problem as a large number of organizations do not have enterprise architectures, although, as recognized in the evaluation, less complex organizations could also create those architectures when needed.

Of the seven steps in the proposed method, four steps need to be executed for each project as they involve project-specific analysis. Three of the steps are related to the complexity analysis. Three other steps are related to other types of analysis that are used in the method: cost, risk, and benefits analysis. These analysis steps are based on the enterprise architectures. The evaluation shows that organizations currently have several different techniques in place to calculate the criteria that they deem necessary, such as costs and risks. Specific modeling techniques and mathematical calculations are included in the descriptions of the specific steps and are based on the literature review from Chapter 2. The focus of this research was not particularly on those other analysis steps such as risks and costs, as these are extensively researched and widely adopted in practice already. The novel element of this research is the complexity analysis, and the application of it in the context of those existing calculation techniques shows that the complexity analysis can be used separately from the rest of the method, and can therefore also be integrated into the existing project portfolio selection processes. This facilitates a more smooth adoption of the complexity analysis in practice. The seventh and last step combines the output of the previous six steps and results in a prioritized list of projects. The calculations in the method can be complicated for some people, and currently need to be executed manually. No software is available yet to calculate those metrics automatically. Therefore, the use of the method is limited to people that can calculate the metrics themselves. The metrics are selected from an existing measurement model for enterprise architecture complexity. The metrics in the complexity analysis are selected by prevalence in literature, but also on their expected variability as a result of a possible change project. A metric is considered relevant when a project under consideration would have a significant influence on the metric. The metric is disregarded when it is not significantly impacted by a single project. This choice is made to keep the method lean and prohibit any unnecessary calculations and steps.

The third and last research question involves the evaluation of the method and is formulated as follows: *How can the proposed method be validated?*. This question is answered in Chapter 4 and Chapter 5. First, the method is applied to a real-life case study that is provided by a company that is specialized in project portfolio management software. The software is implemented at large organizations and (semi) governmental organizations. Fortes, therefore, has a lot of experience with implementing project portfolio management. The case study resulted in a prioritized list of projects and is described in Chapter 4. The model including the results of the case study was presented to five experts at Fortes, and presented in Chapter 5. The five experts were interviewed based on the UTAUT questionnaire. Originally, the UTAUT identified constructs and questions are defined as questions fit for quantitative analysis. In this case and with the available amount of interviewees it was decided to rephrase the questions from the relevant constructs from the UTAUT questionnaire such that qualitative analysis was possible. The transcripts of the interviews were analyzed with the general inductive approach. Categories are identified in the text and prioritized in the results based on their prevalence. The overall results are positive, and state that the method is fit for use in practice, aside from some potential improvement points that are identified in the upcoming sections.

6.2 CONTRIBUTIONS

The contributions of this research are divided into contributions to theory and contributions to practice. The following sections will describe in detail what contributions are identified.

6.2.1 Contributions to Theory

This section describes the contributions of this research to theory. The contributions are as follows.

- In this research, we proposed the integration of Enterprise Architecture Complexity with Project Portfolio management, specifically in the Project Portfolio Selection. Enterprise Architecture Complexity is used as a criterion in the Project Selection Analysis.
- In this research, a novel measurement model for Enterprise Architecture Complexity was designed based on prior research and combined with a rather new MCDA method, the Best-Worst Method. This method is much more user-friendly, as it requires less input. This results in more consistent results in comparison to other MCDA methods, such as AHP.
- In this research, a novel project selection analysis technique is designed to prioritize possible project alternatives by applying the Best-Worst Method. This method is much more user-friendly, as it requires less input. This results in more consistent outcomes in comparison to other MCDA methods, such as AHP (Rezaei, 2015).

6.2.2 Contributions to Practice

This section describes the contributions of this research to practice. The contributions are as follows.

- The designed method can be used by organizations to help select projects based on specific criteria. No project selection method exists yet that includes an Enterprise Architecture Complexity Analysis in the decision criteria.
- The measurement model for enterprise architecture enables dynamic weight measurements for the different metrics, such that the strategic focus of an organization can result in a heavier weight for example for modularity. This makes the method wider applicable.
- The measurement model for enterprise architecture results in a complexity number that can be used in isolation from the rest of the model. Therefore, the calculation of complexity can also be integrated with existing project selection methods that use weighted criteria.
- The designed method uses the Best-Worst Method, which is proven to be easier to use and apply in practice and is also more consistent than other MCDA methods such as AHP, which is widely used in similar project selection methods

6.3 LIMITATIONS

During the execution of this research, several choices have been made that result in several limitations.

- Several assumptions had to be made in regards to the enterprise architecture of Fortes in the case study. Fortes did not have an (up-to-date) Enterprise Architecture. Therefore, the architecture was made during the case study to fulfill the prerequisites of the method. Due to time constraints, the architecture is not as detailed as the situation in practice but captures the essential elements. The depth of the architecture is limited.
- The calculation of the cost of rework in the case study was difficult because it was hard to determine the hidden cost element due to the lack of previous data regarding previous choices in the architecture. The metric was excluded in the case study to prevent possible bias in the results.

- The case study was executed in one company to demonstrate the use of the method in practice. Due to time constraints, no other case studies were investigated. The evaluation in itself also focused on one group of people, namely employees and consultants at Fortes. As the demonstration and evaluation are limited to the Fortes context, we recommend further evaluation in practice.
- The weighting of the criteria in the complexity analysis is quantitative but still based on the judgment of the persons that fill in the method. Objectivity can therefore not be guaranteed, as it opens up some subjectivity in determining the scores. This can partly be solved by determining the weights with multiple persons such that the scores cannot be influenced by someone's preferences. Future research could investigate whether static weights are more suitable, and if that is the case, determine what those weights should be.

6.4 FUTURE WORK

In this section, the future work is described. These are the points that are recognized by the researcher as improvement points for the research. Also, solutions for the limitations in the previous section are mentioned.

- Currently, only objective complexity is considered in the measurement model of Enterprise Architecture Complexity. However, future research could prove if more accurate complexity measurements can be made when also subjective complexity is considered in the measurement model.
- Currently, the Best-Worst Method is used as MCDA in the Complexity Analysis and the Project Selection Analysis. However, also other methods could be considered, such as DEA or other methods listed in Chapter 2.
- Currently, we proposed the integration of Enterprise Architecture Complexity in the Project Portfolio Selection Process. However, Enterprise Architecture Complexity can also be beneficial in other processes in Project Portfolio Management, such as Portfolio Optimization.
- Currently, the measurement model for complexity is quite limited to a few metrics. In future work, a more in-depth analysis of Enterprise Architecture Complexity can result in more accurate measurements.
- Currently, Enterprise Architecture Complexity serves as an input for Project Portfolio Selection. However, when the project is successfully executed, the calculated decrease in complexity can be measured, in comparison to the as-was architecture before the implementation to calculate the realized decrease or increase in complexity. Some kind of score and weight optimization can be investigated to make the measurement model even more accurate in practice.
- Currently, no automation exists for the execution of this method. To make the method easier to use, a program can be developed that incorporates the proposed method. The calculation of Enterprise Architecture Complexity is quite complex. A program that could analyze those metrics from Enterprise Architectures would make the method much simpler to apply. Less knowledge would be required.
- Currently, the measurement model for Enterprise Architecture Complexity uses the number of elements as a metric. During the interviews, it was brought up that more elements would not necessarily lead to linearly more complexity. The effect of more elements could be higher when the architecture consists of more elements, or the effect could be lower. Future research could research this angle.

- During the research evaluation interviews, it was brought up that the model is more suitable for alternative project selection within a specific functional area but less suitable for comparing two fundamentally different projects. The results were inconclusive and denied by another interviewee. However, it is an important issue that should be investigated.
- The project selection analysis now weights all criteria equally. In the case an organization wants to value the complexity higher than for example the cost, it is now not possible. Adding an MCDA weighting step for each criterion would make it possible to weigh the criteria differently.

7 REFERENCES

Aldea, A., Iacob, M.-E., Quartel, D., "From Business Strategy to Enterprise Architecture and Back," 22nd International Enterprise Distributed Object Computing Workshops, 2018, pp. 145-152.

Aldea, A., Iacob, M.-E., Daneva, M., & Masyhur, L. H. (2019). Multi-criteria and model-based analysis for project selection: An integration of capability-based planning, project portfolio management, and enterprise architecture. Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOCW, 2019-Octob, 128–135. https://doi.org/10.1109/EDOCW

Alwadain, A., Fielt, E., Korthaus, A., Rosemann, M., "A critical realist perspective of Enterprise Architecture evolution: Conditioning and outcomes," Australasian Journal of IS, vol. 18, no. 3, 2014.

Al-Harbi, K. M. A.-S. (2001). Application of the AHP in project management. International Journal of Project Management, 19(1), 19–27. https://doi.org/https://doi.org/10.1016/S0263-7863(99)00038-1

Anderson, C., Theraulaz, G., and Deneubourg, J.L. 2002. “Self-assemblages in Insect Societies,” Insectes Societies

Antlova, K. (2010). Critical success factors for the implementation of ICT projects. In Communications in Computer and Information Science: Vol. 109 CCIS (Issue PART 1). https://doi.org/10.1007/978-3-642-16402-6_17

Anyosa Soca, V., “Linking portfolio, program, and projects to business strategy,” 2009.

Archer, N. P., & Ghasemzadeh, F. (1999). An integrated framework for Project Portfolio Selection. International Journal of Project Management, 17(4), 207-216.

Arthur, B.W. 1994. “Inductive Reasoning and Bounded Rationality,” American Economic Review (84), pp. 406–411.

Bachmann, P. 1894. Die Analytische Zahlentheorie. Leipzig, Germany: Teubner.

Beese, J., Aier, S., Haki, M. K., & Aleatrati Khosroshahi, P. (2016). Drivers and effects of information systems architecture complexity: A mixed-methods study. 24th European Conference on Information Systems, ECIS 2016, June.

Bhattacharya, P. (2018). Aligning Enterprise Systems Capabilities with Business Strategy: An extension of the Strategic Alignment Model (SAM) using Enterprise Architecture. Procedia Computer Science, 138, 655–662. doi:10.1016/j.procs.2018.10.087

Bonchev, D., and Buck, G.A. 2005. “Quantitative Measures of Network Complexity,” Complexity in Chemistry, Biology, and Ecology, pp. 191–235.

Breu, R., Innerhofer-Oberperfler, F., & Yautsiukhin, A. (2008). Quantitative Assessment of Enterprise Security System. 2008 Third International Conference on Availability, Reliability, and Security, 921–928. https://doi.org/10.1109/ARES.2008.164

Cooper, R., Edgett, S., & Kleinschmidt, E. (2001). Portfolio management for new product development: results of an industry practices study. r&D Management, 31(4), 361- 380.

Costa, J., Vasconcelos, A., & Fragoso, B. (2020). An enterprise architecture approach for assessing the alignment between projects and goals. International Journal of Information Technology Project Management, 11(3), 55–76. https://doi.org/10.4018/IJITPM.2020070104

Danesh, D., Ryan, M. J., & Abbasi, A. (2015). Using Analytic Hierarchy Process as a Decision-Making Tool in Project Portfolio Management. World Academy of Science, Engineering and Technology, International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering, 9(12), 4163-4173.

Daniels, C. B. & LaMarsh II, W. J. (2007). Complexity as a cause of failure in information technology project management. In IEEE International Conference on System of Systems Engineering, (pp. 1– 7)., San Antonio, TX.

Davis, J. S. & LeBlanc, R. J. (1988). A study of the applicability of complexity measures. IEEETransaction on Software Engineering, 14(9), 1366–1372.

Elbok, G., & Berrado, A. (2017). Towards an effective Project Portfolio Selection process. Proceedings of the International Conference on Industrial Engineering and Operations Management, 2158–2169.

Fioretti, G. 1999. "A Subjective Measure of Complexity," Advances in Complex Systems (2:4), pp. 349–370.

Flückiger, M., and Rauterberg, M. 1995. Komplexität und Messung von Komplexität. Technical Report IfAP/ETH/CC-01/95, ETH Zurich.

Forrester, J.W. 1961. Industrial Dynamics. Portland: Productivity Press. Frese,

Fritscher, B., & Pigneur, Y. (2011). Business IT alignment from business model to Enterprise Architecture. Paper presented at the International Conference on Advanced Information Systems Engineering.

Garcia, I., "Alignment between an Organization's Projects and Strategic Objectives," Inst. Super. Técnico, 2017.

Gardner, M. 1970. "Mathematical Games – The Fantastic Combinations of John Conway's New Solitaire Game "Life"," Scientific American, 223, pp. 120–123.

Gellweiler, C. (2020). Connecting enterprise architecture and project portfolio management: A review and a model for IT project alignment. International Journal of Information Technology Project Management, 11(1), 99–114. https://doi.org/10.4018/IJITPM.2020010106

Ghasemzadeh, F., & Archer, N. P. (2000). Project portfolio selection through decision support. Decision support systems, 29(1), 73-88.

Griffioen, A., & Hofman, F. (2012). Choosing a language and tool for the Business and Information Architecture of Statistics Netherlands.

Helfert, M., Doucek, P., & Maryska, M. (2013). The "enterprise architect" — A new approach to business informatics management. Quality Innovation Prosperity, XVII(1), 67–87.

Henderson, J. C., & Venkatraman, N. (1993). Strategic alignment: Leveraging information technology for transforming organizations. IBM Systems Journal, 32(1), 4–16. doi:10.1147/sj.382.0472

Henningsson, S., Hanseth, O., (2011). The Essential Dynamics of Information Infrastructures. International Conference on Information Systems 2011, ICIS 2011. 3.

Ishizaka, A., & Labib, A. (2011). Review of the main developments in the analytic hierarchy process. Expert systems with applications, 38(11), 14336-14345.

Ishizaka, A., & Nemery, P. (2013). Multi-criteria decision analysis: methods and software: John Wiley & Sons.

ISO/IEC/IEEE (2011). ISO/IEC/IEEE 42010: Systems and software engineering - Architecture description. Technical report, International Organization for Standardization.

Kauffman, S. 1996. At Home in the Universe: The Search for the Laws of Self-Organization and Complexity. Oxford University Press.

Khosroshahi, P. A., Beese, J., & Aier, S. (2016). What Drives Application Portfolio Complexity? An Empirical Analysis of Application Portfolio Cost Drivers at a Global Automotive Company. Proceedings - CBI 2016: 18th IEEE Conference on Business Informatics, 1, 282–289. https://doi.org/10.1109/CBI.2016.39

Kitchenham, B. & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical report, Evidence-Based Software Engineering Project, Durham

Kolmogorov, A. 1998. "On Tables of Random Numbers," Theoretical Computer Science, 207(2), pp. 387– 395.

Landauer, R. 1988. "A Simple Measure of Complexity," Nature, 336, pp. 306–307.

Langermeier, M., & Bauer, B. (2018). A model-based method for the evaluation of project proposal compliance within EA planning. Proceedings of the 22nd International Enterprise Distributed Object Computing Workshop, Stockholm, Sweden, (EDOC 2018) (pp. 97–106). Academic Press. doi:10.1109/EDOCW.2018.00024

Lankhorst, M. (2017). Enterprise architecture at work. Berlin: Springer.

Lucke, C., Krell, S., & Lechner, U. (2010). Critical issues in enterprise architecting: A literature review. In Americas Conference on Information Systems, Lima.

Malta, P. M., Sousa, R. D., "Looking for effective ways of achieving and sustaining Business-IT alignment," 5th Iberian Conference on Information Systems and Technologies, 2010, pp. 1-5.

Mosthaf, J., & Wagner, H.-T. (2016). The Architect's Role in Business-IT Alignment. International Journal of IT/Business Alignment and Governance, 7(1), 36–49. doi:10.4018/IJITBAG.2016010103

Niemi, E. (2007). Enterprise architecture stakeholders A holistic view. In Americas Conference on Information Systems, Keystone, CO.

Obitz, T., Babu, M., "Enterprise architecture expands its role in strategic business transformation: Infosys Enterprise Architecture survey 2008/2009," Infosys, Infosys report 2009.

Otra-Aho, Ville & Iden, Jon & Hallikas, Jukka. (2019). The Impact of the Project Management Office Roles to Organizational Value Contribution. International Journal of Information Technology Project Management. 10. 79-99. 10.4018/IJITPM.2019100103.

Padovani, M., & Carvalho, M. M. (2016). Integrated PPM Process: Scale Development and Validation. International Journal of Project Management, 34(4), 627-642.

Peffers, K., Tuunanen, T., Rothenberger, M. A., Chatterjee, S., "A Design Science Research Methodology for Information Systems Research," J. Manag. Inf. Syst., vol. 24, no. 3, pp. 45–77, 2007.

PMI. (2013). The standard for portfolio management (3rd ed.). Newtown Square, PA: Project Management Institute.

Rezaei, J. (2015). Best-worst multi-criteria decision-making method. Omega (United Kingdom), 53, 49–57. https://doi.org/10.1016/j.omega.2014.11.009

Rosen, R. 1977. "Complexity as a System Property," International Journal of General Systems, 3(4), pp. 227–232.

Russell, J. S., & Skibniewski, M. (1987). A structured approach to the contractor prequalification process in the USA. CIB-SBI Fourth Int. Sym. on Building Economics, Session D, 240-51.

Schneberger, S.L, McLean, E.R, The Complexity Cross: Implications for Practice. Communications of the ACM, 46 (9), 216- 225, 2003.

Shannon, C.E., and Weaver, W. 1949. The Mathematical Theory of Communication. The University of Illinois Press, Urbana, Illinois.

Slot, R., Dedene, G., Maes, R.: Business Value of Solution Architecture. In: Proper, E., Harmsen, L., Dietz, J.L.G. (eds.) Proc. First NAF Academy Working Conference on Practice-Driven Research on Enterprise Transformation, PRET 2009, held at CAiSE 2009, Amsterdam, The Netherlands, June 11. LNBIP, vol. 28, pp. 84–108 (2009)

Sousa, P., & Carvalho, M. (2018). Dynamic organization's representation. Linking project management with Enterprise Architecture. Proceeding 2018 20th IEEE International Conference on Business Informatics, CBI 2018, 2, 170–174. https://doi.org/10.1109/CBI.2018.10066

Sterman, J.D. 2000. Business Dynamics – Systems Thinking and Modeling for a Complex World. McGraw-Hill Higher Education.

Strano, C., & Rehmani, Q. (2007). The role of the enterprise architect. Information Systems and e-Business, 5(4), 379–396.

The Open Group (2011). TOGAFVersion 9.1. U.S.: The Open Group.

The Open Group. (2016). ArchiMate® 3.0 Specification: Van Haren.

Thomas, David. (2006). A General Inductive Approach for Analyzing Qualitative Evaluation Data. American Journal of Evaluation. 27. 237-246. 10.1177/1098214005283748.

Ullah, A., & Lai, R. (2013). A systematic review of business and information technology alignment. ACM Transactions on Management Information Systems, 4(1), 4. doi:10.1145/2445560.2445564

Venkatesh, V., Morris, M. G., Davis, G. B., Davis, F. D., "User acceptance of information technology: Toward a unified view," MIS Quarterly, pp. 425-478, 2003.

Weaver, W. 1948. "Science and Complexity," American Scientist (36), pp. 536–544.

8 APPENDIX

8.1 LIST OF SOURCES FROM SLR

1	Aldea, A., Iacob, M.-E., Daneva, M., & Masyhur, L. H. (2019). Multi-criteria and model-based analysis for project selection: An integration of capability-based planning, project portfolio management, and Enterprise Architecture. *Proceedings IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, *2019-Octob*, 128–135. https://doi.org/10.1109/EDOCW
2	Antlova, K. (2010). Critical success factors for the implementation of ICT projects. In *Communications in Computer and Information Science: Vol. 109 CCIS* (Issue PART 1). https://doi.org/10.1007/978-3-642-16402-6_17
3	Bartenschlager, J., & Goeken, M. (2010). IT strategy implementation framework Bridging Enterprise Architecture and IT governance. *16th Americas Conference on Information Systems 2010, AMCIS 2010*, *1*, 379–388.
4	Beese, J., Aier, S., Haki, M. K., & Aleatrati Khosroshahi, P. (2016). Drivers and effects of information systems architecture complexity: A mixed-methods study. *24th European Conference on Information Systems, ECIS 2016*, *June*.
5	Bhattacharya, P. (2017). Modelling Strategic Alignment of Business and IT through Enterprise Architecture: Augmenting ArchiMate with BMM. *Procedia Computer Science*, *121*, 80–88. https://doi.org/10.1016/j.procs.2017.11.012
6	Bhattacharya, P. (2018). Aligning enterprise systems capabilities with business strategy: An extension of the Strategic Alignment Model (SAM) using Enterprise Architecture. *Procedia Computer Science*, *138*, 655–662. https://doi.org/10.1016/j.procs.2018.10.087
7	Bouafia, K., & Molnár, B. (2019). Analysis approach for enterprise information systems architecture based on hypergraph to aligned business process requirements. *Procedia Computer Science*, *164*, 19–24. https://doi.org/10.1016/j.procs.2019.12.149
8	Bucher, D. A., & Min, I. A. (2017). Five pillars of enterprise portfolio planning. *IEEE Aerospace Conference Proceedings*. https://doi.org/10.1109/AERO.2017.7943945
9	Buckl, S., Ernst, A. M., Matthes, F., & Schweda, C. M. (2009). Visual Roadmaps for Managed Enterprise Architecture Evolution. 2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 352–357. https://doi.org/10.1109/SNPD.2009.99
10	Chikofsky, E. (2006). Sustain, enhance, or replace: Making decisions on systems. *IEEE International Conference on Software Maintenance, ICSM*, 134. https://doi.org/10.1109/ICSM.2006.65
11	Costa, J., Vasconcelos, A., & Fragoso, B. (2020). An Enterprise Architecture approach for assessing the alignment between projects and goals. *International Journal of Information Technology Project Management*, *11*(3), 55–76. https://doi.org/10.4018/IJITPM.2020070104
12	Elbok, G., & Berrado, A. (2017). Towards an effective Project Portfolio Selection process. *Proceedings of the International Conference on Industrial Engineering and Operations Management*, 2158–2169.
13	Gellweiler, C. (2020). Connecting Enterprise Architecture and project portfolio management: A review and a model for IT project alignment. *International Journal of Information Technology Project Management*, *11*(1), 99–114. https://doi.org/10.4018/IJITPM.2020010106

14	Iacob, M. E., Monteban, J., Van Sinderen, M., Hegeman, E., & Bitaraf, K. (2018). Measuring Enterprise Architecture Complexity. *Proceedings IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, *2018-Octob*, 115–124. https://doi.org/10.1109/EDOCW.2018.00026
15	Iamratanakul, S., Patanakul, P., & Milosevic, D. (2008). Project portfolio selection: From past to present. Proceedings of the 4th IEEE International Conference on Management of Innovation and Technology, ICMIT, 287–292. https://doi.org/10.1109/ICMIT.2008.4654378
16	Kaisler, S. H., Armour, F., & Valivullah, M. (2005). Enterprise Architecting: Critical Problems. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 224b-224b. https://doi.org/10.1109/HICSS.2005.241
17	Khosroshahi, P. A., Beese, J., & Aier, S. (2016). What Drives Application Portfolio Complexity? An Empirical Analysis of Application Portfolio Cost Drivers at a Global Automotive Company. *Proceedings CBI 2016: 18th IEEE Conference on Business Informatics*, *1*, 282–289. https://doi.org/10.1109/CBI.2016.39
18	Khosroshahi, P. A., Matthes, F., Beese, J., Winter, R., & Yilmaz, F. (2017). Key Performance indicators for a capability-based application portfolio management. *Proceedings IEEE International Enterprise Distributed Object Computing Workshop, EDOCW*, *2017-Octob*, 85–91. https://doi.org/10.1109/EDOCW.2017.22
19	Lakhrouit, J., & Baïna, K. (2015). Evaluating complexity of Enterprise Architecture components landscapes. *2015 10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015*, 0–4. https://doi.org/10.1109/SITA.2015.7358443
20	Langermeier, M., & Bauer, B. (2018). A Model-Based Method for the Evaluation of Project Proposal Compliance within EA Planning. *2018 IEEE 22nd International Enterprise Distributed Object Computing Workshop (EDOCW)*, 97–106. https://doi.org/10.1109/EDOCW.2018.00024
21	Lankhorst, M. M. M. M., Quartel, D. A. C. D. A. C., & Steen, M. W. A. M. W. A. (2010). Architecture-Based IT Portfolio Valuation. In *Lecture Notes in Business Information Processing: Vol. 69 LNBIP* (pp. 78–106). https://doi.org/10.1007/978-3-642-16770-6_4
22	Lehnert, M., Linhart, A., Manderscheid, J., & Svechla, M. (2016). V3PM: A decision support tool for value-based process project portfolio management. *24th European Conference on Information Systems, ECIS 2016*.
23	Maciaszek, L. A. (2006). Managing complexity of enterprise information systems. In J. Seruca, I and Cordeiro, J and Hammoudi, S and Filipe (Ed.), *Enterprise Information Systems VI* (pp. 30–36). SPRINGER. https://doi.org/10.1007/1-4020-3675-2_4
24	Quartel, D., Steen, M. W. A., & Lankhorst, M. (2010). IT Portfolio Valuation. *2010 14TH IEEE INTERNATIONAL ENTERPRISE DISTRIBUTED OBJECT COMPUTING CONFERENCE (EDOC 2010)*, 3–13. https://doi.org/10.1109/EDOC.2010.24
25	Quartel, D., Steen, M. W. A., & Lankhorst, M. M. (2012). Application and project portfolio valuation using Enterprise Architecture and business requirements modelling. *Enterprise Information Systems*, *6*(2), 189–213. https://doi.org/10.1080/17517575.2011.625571
26	Rodrigues, L. S., Amaral, L., & Minho, U. (2016). Key dimensions of the Enterprise Architecture value drivers: Time frame and quantification. *Proceedings of the 28th International Business Information Management Association Conference Vision 2020: Innovation Management, Development Sustainability, and Competitive Economic Growth*, 2552–2554.
27	Schneider, A. W., Zec, M., & Matthes, F. (2014). Adopting notions of complexity for Enterprise Architecture management. 20th Americas Conference on Information Systems, AMCIS 2014, July.

28	Simon, D., Fischbach, K., & Schoder, D. (2013). Integrating IT Portfolio Management with Enterprise Architecture Management. *Enterprise Modelling and Information Systems Architectures*, *8*(2), 79–104. https://doi.org/10.1007/s40786-013-0005-9
29	van den Berg, M., Slot, R., van Steenbergen, M., Faasse, P., & van Vliet, H. (2019). How Enterprise Architecture improves the quality of IT investment decisions. *Journal of Systems and Software*, *152*, 134–150. https://doi.org/10.1016/j.jss.2019.02.053
30	Wehling, K., Wille, D., Seidl, C., & Schaefer, I. (2017). Decision support for reducing unnecessary IT complexity of application architectures. *Proceedings 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceedings*, 161–168. https://doi.org/10.1109/ICSAW.2017.47
31	Zhang, M., Chen, H., & Luo, A. (2018). A Systematic Review of Business-IT Alignment Research with Enterprise Architecture. *IEEE Access*, *6*, 18933–18944. https://doi.org/10.1109/ACCESS.2018.2819185

Table 36. SLR articles

8.2 MATRIX RESULT FOR CONTENT OF ARTICLES FROM SLR

REF	PPM	PS	PPM EA	EA	EA PS	EAC	EACM
1	x	x	x	x	x		
2	x						
3	x		x	x			
4				x		x	x
5	x		x	x			
6	x		x	x			
7				x			
8	x						
9				x			
10	x	x	x	x	x		
11	x	x	x	x	x	x	x
12	x						
13	x	x					
14	x	x	x	x	x		
15	x			x		x	x
16				x			
17	x		x	x	x	x	x
18	x		x	x	x	x	x
19				x	x	x	
20	x	x	x	x	x		
21	x		x	x	x		
22	x	x					
23	x		x	x		x	x
24	x		x	x	x	x	x
25	x		x	x			
26				x			
27				x		x	
28	x		x	x	x		
29	x		x	x	x		
30	x		x	x	x		
31	x		x	x		x	

Table 37. Concept matrix for SLR

8.3 List of Definitions

Term	Definition
Project	A temporary endeavor that is undertaken to create a unique product, service, or result.
Program	A group of related projects, subprograms, and program activities that are managed in a coordinated way to obtain benefits not available from managing them.
Portfolio	A group of related projects, subprograms, and program activities that are managed in a coordinated way to obtain benefits not available from managing them
Project Management	The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements
Program Management	The application of knowledge, skills, tools, and techniques to a program to meet the program requirements and to obtain benefits and control not available by managing projects individually.
Portfolio Management	The centralized management of one or more portfolios to achieve strategic objectives.
Architecture	The fundamental concepts or properties of a system in its environment are embodied in its elements, relationships, and the principles of its design and evolution (ISO/IEC/IEEE, 2011).
Enterprise	Any collection of organizations that has a common set of goals (The Open Group, 2011).
Enterprise Architecture	A coherent whole of principles, methods, and models used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure (Lankhorst et al. 2013).
Complexity	a property with a measurable value based on metrics that are relevant for the aspect under consideration

Table 38. List of Definitions

8.4 ARCHIMATE

The ArchiMate language is considered the best and the most common language for developing architecture. Griffioen and Hofman (2012) argued in their research that the reason to choose ArchiMate above other formal languages in EA as a language was because it is an open standard. Moreover, ArchiMate is particularly attractive, mainly due to the usage of visual representation, and the fact that it encourages the use of colors to highlight the different modeled layers. ArchiMate also opts for one unique language (UML) to model every layer of the architecture, which eases the communication when teams responsible for the different layers need to collaborate (Fritscher & Pigneur, 2011). It offers an integrated architectural approach that describes and visualizes different architecture domains and their underlying relations and dependencies. Its language framework provides a structuring mechanism for architecture domains, layers, and aspects (The Open Group, 2016).

The ArchiMate language consists of different aspects and layers that together make up a framework known as ArchiMate Core Framework (The Open Group, 2016). The dimensions of the core framework of ArchiMate are explained as follows:

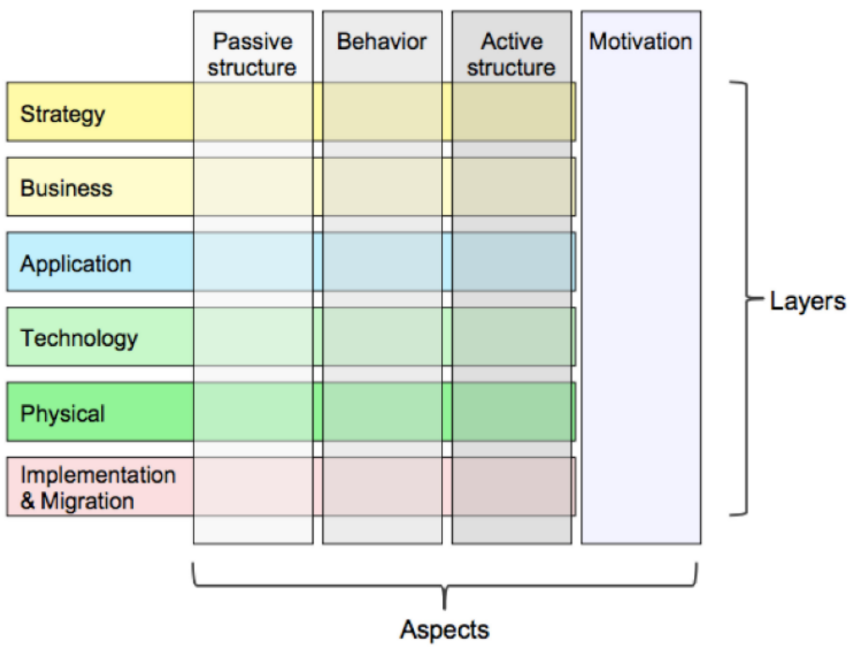


Fig 30. Full ArchiMate Framework (The Open Group, 2016)

The three levels at which an enterprise can be modeled in ArchiMate are the business layer, the application layer, and the technology layer. The business layer depicts business services offered to customers, which are realized in the organization by business processes performed by business actors. The application layer depicts application services that support the business, and the applications that realize them. The technology depicts technology services such as processing, storage, and communication services needed to run the applications, and the computer and communication hardware and system software that realize those services.

The three aspects at which an enterprise can be modeled in ArchiMate are the active structure aspect, the behavior aspect, and the passive structure aspect. The active structure aspect, which represents the structural elements (the business actors, application components, and devices that display actual behavior; i.e., the "subjects" of activity). The behavior aspect, which represents the behavior (processes, functions, events, and services) performed by the actors. Structural elements are assigned to behavioral elements, to show who or what displays the behavior. The passive structure aspect, which represents the objects on which behavior is performed. These are usually information objects in the Business Layer and data objects in the Application Layer, but they may also be used to represent physical objects.

In the recent development of ArchiMate language, ArchiMate 3.0 specification by The Open Group (2016), three layers and one aspect were added to the framework. Firstly, the physical elements were built upon the technology later to add elements for physical facilities and equipment, distribution networks, and materials. The motivation aspect was also introduced at a generic level to model the motivations or reasons that guide the design or change of an Enterprise Architecture. Lastly, an implementation and migration layer was also added for architectural elements that are related to implementation processes such as deliverables, work packages, etc. The summary of the concept in the full ArchiMate framework based on ArchiMate 3.0 specification is presented in Figure 5.

As can be seen in Figure 5, ArchiMate consists of many blocks as a combination of different layers and aspects. Since we want to use ArchiMate for modeling, it is important to note that in modeling we need the concept of elements and relationships. An element could be part of any of the blocks in the framework above (e.g. a behavior element or motivation element), while a relationship connects a source and target concept (could be either element or other relationship). A complete explanation regarding the core elements and relationships along with their definitions is provided in the book by The Open Group (2016). In the next section, some related analysis needed along with the appropriate modeling concept in ArchiMate will be explained with the purpose to define the theoretical framework required for this study.

8.5 EA COMPLEXITY DIMENSIONS

The four dimensions of Schneider, Zec, and Matthes (2014) are adopted in this research. This section elaborates more on those dimensions. Each subsection describes a dimension as defined by Schneider, Zec, and Matthes (2014).

Organized complexity versus disorganized complexity

The first dimension is based on the number of variables to be considered and their relations (Weaver, 1948). According to Weaver, disorganized complexity is a result of a large number of parts. This could be millions or even billions. The overall interactions between those parts appear to be random but can be analyzed by using probability theory and statistical methods. An example of organized complexity can be all the people in the world. They appear to behave differently, yet it is possible to apply statistical methods to analyze the behavior of the group as a whole.

Organized complexity is the opposite; non-random, or correlated interactions between those parts. This structure of parts can interact with other structures or systems. The system has properties that are or cannot be expressed by the individual parts. There is no necessity for a large number of parts for the system to have emergent behavior. The system can be modeled and simulated. An example of organized complexity is a smart city, of which its residents are parts of the system.

Qualitative complexity versus quantitative complexity

The second dimension distinguishes between quality and quantity. Thereby, qualitative complexity refers to the qualitative evaluation of a certain attribute of variables or a system. An example is the "El Farol Problem" (Arthur, 1994). In this multiple-stage game, participants have to decide whether to visit a bar or not in each round. They all prefer to enjoy a drink at the bar rather than staying at home, but the bar has a maximum capacity of seats. Of course, it is less enjoyable to attend an overcrowded bar than staying at home. For each round, it can be determined whether a participant does attend the bar or not and whether he is better off doing so. Remembering the decisions on the other hand will not provide new insights for the next round since the participants´ decisions might change in each round. This game is considered to be a complex problem that needs to be solved by every participant every round. The value of a decision depends on the decisions of all other players. This type of complexity is independent of the number of players, the number of rounds, or the memory capacity of players. Researchers studying complex system phenomena use a qualitative notion of complexity as well, such as self-organization (Kauffman, 1996), emergence (Anderson, 2002), or dynamical systems (Gardner, 1970).

Other researchers apply a quantitative notion of complexity. Kolmogorov (1998) proposed a classic measure of quantitative complexity. The Kolmogorov complexity is the length of the shortest computer program capable of generating a given string. Another fundamental quantitative measure to which many complexity measures relate is entropy (Shannon and Weaver, 1949), which can be understood as a measure for uncertainty in a message. Other approaches have been developed to measure (computing) complexity as well, for instance, based on the number and variety of both components and their interactions within a system (Schneberger et al., 2003). Quantitative measures suggest that the quantity of a particular property directly influences complexity. For instance, computer scientists describe the complexity of algorithms as a function of the input length. Algorithms are classified according to their asymptotic behavior for large inputs using Landau notation (Bachmann, 1894). Typically, the number of calculations or the amount of memory consumption is of interest to determine algorithmic complexity.

Subjective complexity versus objective complexity

The third dimension of complexity is based on the role of the observer. Objective complexity refers to a notion of complexity that is independent of the observer. Complexity is considered to be a property of the system under observation, much in the same way as the mass or volume of a physical body (Fioretti, 1999). Such objective views are prevalent, for example, in the domain of qualitative complexity where system properties like emergence (Anderson, 2002) are investigated. The same applies to most of the developed complexity metrics as their results are free of individual influence (Landauer, 1988).

However, complexity can also be considered to be a property of the relationship between a system and its observer (Rosen, 1977). Thereby, the observer will perceive a system as complex if his/her mental model of the system cannot explain his/her observations. In contrast to the objective complexity notion, subjective complexity is bound to the existence of an individual observing a system. Researchers define subjective measures, for instance, based on mental categories of the observer (Fioretti, 1999) or as being composed of other objective measures (Flückiger et al., 1995).

Structural complexity versus dynamic complexity

One pole of the fourth dimension is known as structural complexity, which is also known as combinatorial or detail complexity (Sterman, 2000). It covers a pattern of system components, i.e. the number of variables as well as the cause-and-effect relationships between them. A structural perspective is employed, for example, in network research where cyclic groups, spanning sub-graphs, and extended connectivity play an important role (Bonchev et al., 2005). The two well-known measures of complexity, i.e. Kolmogorov complexity (Kolmogorov, 1998) and entropy (Shannon et al., 1949), also apply a structural notion of complexity.

In contrast, dynamic complexity refers to the observation of the multifaceted interdependencies as well as changes of interactions between variables of a system. Therefore, "dynamic complexity arises from the interactions among the agents over time" (Sterman, 2000). In complex systems, the impact of actions often cannot be reversed. Therefore a comparison between system states in the past and the current one is rather difficult. With several interacting feedbacks, determining an exclusive effect of a certain variable is hardly possible since it is likely that other variables change as well. As a consequence, the system behavior interpretation is usually complicated. Additionally, delays in cause and effect have to be considered, which can result in system instability and influence the dynamics of a complex system. Dynamic complexity arises, for example, when systems are strongly interacting with each other and the natural world or if actions influence future choice options (Sterman, 2000). The dynamic complexity notion has also been applied in socio-economics (Forrester, 1961).

8.6 SLR Enterprise Architecture Complexity Metrics

Metric	Objective/subjective	Structural/dynamic	Quantitative/qualitative	Ordered/disordered
# relations	Objective	Structural	Quantitative	Ordered
# elements	Objective	Structural	Quantitative	Ordered
# cardinal elements	Objective	Structural	Quantitative	Ordered
# cardinal relations	Objective	Structural	Quantitative	Ordered
Cyclomatic complexity	Objective	Structural	Quantitative	Ordered
Element entropy	Objective	Structural	Quantitative	Disordered
Relation entropy	Objective	Structural	Quantitative	Disordered
Conformity	Objective	Structural	Quantitative, qualitative	Disordered
Interface Complexity Multiplier	Objective	Structural, dynamic	Quantitative, qualitative	Ordered, disordered
Redundancy	Objective	Structural	Quantitative, qualitative	Disordered
# OS & middleware	Objective	Structural	Quantitative	Ordered
Functions/system	Objective	Structural	Quantitative	Ordered
# patterns	Objective	Structural	Quantitative	Ordered
Application age	Objective	Structural	Quantitative	Ordered
# hardware platforms	Objective	Structural	Quantitative	Ordered
Betweenness centrality	Objective	Structural	Quantitative	Disordered
Quantified expert opinion	Subjective	Structural	Quantitative	Ordered
Pattern coverage	Objective	Structural	Quantitative	Ordered
Elements/type	Objective	Structural	Quantitative	Ordered
Relations/element	Objective	Structural	Quantitative	Ordered
Processes/element	Objective	Structural	Quantitative	Ordered
Elements/process	Objective	Structural	Quantitative	Ordered
Service-time Actual	Objective	Structural	Quantitative	Ordered
Domains/application	Objective	Structural	Quantitative	Ordered
Software categories/app	Objective	Structural	Quantitative	Ordered
SLOC	Objective	Structural	Quantitative	Ordered
Halstead difficulty	Objective	Structural	Quantitative	Ordered
# functions	Objective	Structural	Quantitative	Ordered
Apps/user	Objective	Structural	Quantitative	Ordered
Customization	Objective	Structural	Quantitative	Ordered
# instances	Objective	Structural	Quantitative	Ordered
# software platforms	Objective	Structural	Quantitative	Ordered
Application type	Objective	Structural	Qualitative	Ordered
# software frameworks	Objective	Structural	Quantitative	Ordered
# new applications	Objective	Structural	Quantitative	Ordered
# retired applications	Objective	Structural	Quantitative	Ordered
# physical servers	Objective	Structural	Quantitative	Ordered
# virtual servers	Objective	Structural	Quantitative	Ordered
Visibility Fan-In	Objective	Structural	Quantitative	Ordered
Visibility Fan-out	Objective	Structural	Quantitative	Ordered
Requirements/app	Objective	Structural	Quantitative	Ordered
Propagation cost	Objective	Structural	Quantitative	Disordered

Table 39. Identified metrics and their classification (Iacob, Monteban, van Sinderen, 2018)

8.7 Code Sample for Modularity Calculation in R

This example is based on the case study to-be architecture of project 1. This graph is simple, and with modularity 0, as the architecture is indivisible.

```
# Import the library to use igraph

library(igraph)


# Define all relations between the elements. Start with numbering the elements.

# Then, list the relations as defined below

g <- graph_from_literal( 1--2, 2--3, 2--4, 2--5, 2--6, 2--7, 2--8, 2--9, 2--10, 2--11, 2--12, 2--13, 2--14, 2-
-15 )


# Not necessary to define a layout, but it helps to make the plotted graph easier to understand

g$layout <- layout_on_grid(g, width = 0, height = 0, dim = 2)


# Plot the graph

plot(g)


# Clusterize the graph to calculate the modularity

wtc <- cluster_walktrap(g)


# Output the modularity score for the given graph

modularity(wtc)
```