Master Thesis
Software Engineering
Thesis no: MSE-2010-34
October 2010

Applying Multi-Criteria Decision Analysis for Software Quality Assessment

- Systematic Review and Evaluation of Alternative MCDA Methods

Wan Ai Goh
810216-P018

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 28 weeks of full time studies.

Contact Information:
Author:
Wan Ai Goh
E-mail: wanai_2000@yahoo.com

University advisor(s):	EMSE Supervisor:
Dr. Tony Gorschek	Prof. Dr. H. Dieter Rombach (Technical University of Kaiserslautern)
School of Computing	Dr. Adam Trendowicz (Fraunhofer IESE)

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

ABSTRACT

With the rapid advancement of technologies, software is gaining its popularity in assisting our daily activities in the last decades. This circumstance causes a rising concerns about a software product with high quality which lead to a question about the justification whether a software product has high quality. Therefore, a numerous of researches and studies had spent a lot of effort in software product quality assessment in order to justify whether the software product(s) under study have satisfactory quality. One of the foremost approaches to assess software product quality is the application of the quality models. For example, quality model ISO 9126. However, the quality models do not provide an explicit way to aggregate the performance of different quality aspects nor handling the various interests raised from different perspective or stakeholders.

Although many studies have been conducted to aggregate the different measures of quality attributes, they are still not capable to include the various interests raised by different software product stakeholders. Therefore, some studies have attempted to apply MCDA methods in order to aggregate the measure of quality attributes as the ultimate software product quality and handling the various quality interests. However, they do not provide any rational about their particular choice of MCDA methods. Most of them justify their choice by referring to high popularity of the selected MCDA method. Without studying the suitability of MCDA methods in the application domain of the software product, it is difficult to conclude whether the chosen MCDA methods fit in the intended software engineering discipline. Furthermore, there is no systematic approach available to help other software practitioners in selecting the MCDA method that will be suitable for their needs and constraints in software product quality assessment.

This thesis aims to provide the key concepts for an effective selection of suitable MCDA method for the purpose of software product quality assessment. A foremost part of this thesis presents two systematic reviews. The first review illustrates the evaluation of the characteristics of MCDA methods. The second review identifies the major needs and constraints of the software quality assessment potential MCDA method has to consider in order to be used for assessing quality of software products.

Based on the results from both systematic reviews, a selection framework named MCDA-SQA framework is formulated. This framework is intended to assist the software practitioners to systematically select and adapt appropriate MCDA method(s) in order to fulfil their quality assessment needs and the respective environmental concerns.

Keywords: software quality assessment, multi-criteria decision analysis, MCDA, systematic review, method selection framework

ACKNOWLEDGEMENT

This thesis would not have been possible without the sincere help from several people. I would like to take this opportunity for expressing my sincere gratitude to all of them.

First of all, I would like to my supervisor Dr. Adam Trendowicz, for providing me his precious guidance, support and advices throughout the course of this thesis. His valuable comments and suggestions made the road to achieve our goals smooth and pleasant. I am also thankful to my EMSE co-supervisor Dr. Tony Gorschek from Blekinge Institute of Technology (BTH) for his support and advice that lead to this dissertation.

I am deeply thankful to my family members for their enduring belief in me. Their support has kept me going and enabled me to accomplish this thesis work.

I also would like to thank all my friends, in particular Cheng Chow Kian, for the fruitful discussions, support and suggestion during the course of this thesis work.

Last but not least, I would like to thank Fraunhofer IESE Kaiserslautern to allow me working on my thesis there in the first place and provide the necessary facilities and apparatus throughout my thesis work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Motivation

With the increasing reliance of software product in our daily activities and the industries, the software users are usually demanding high quality of software product [15 - 21]. They are willing to purchase a software product with better and better quality but higher price in order to facilitate their works in a more effective manner. As a result, the software product quality assessment has gained major concerns in the area of software engineering [17]. In this thesis we focus on the method to assess software product quality and exclude other elements of software quality assurance such as, for example, quality assurance activities and models. For brevity, software product quality is referred to as software quality throughout this thesis document.

Software quality is a crucial and hotly debated topic in software engineering because of its definition is highly subjective and abstract [17]. Kitchenham and Walker [21] have recognized the meaning of software quality can be perceived differently as a result from five perspectives which are introduced by David Garvin [16] namely transcendental perspective, user perspective, manufacturing perspective, product-based perspective, and value-based perspective. Nevertheless, the definition of software quality can be generalized to achieve one of these two distinct categories, i.e. the conformance of software product to the predefined requirement specification and the extent a software product satisfies the user expectation [16].

In line with the multiple perspectives about the meaning of software quality, different quality models [4, 5, 10-13] have been introduced to define and measure software quality from product-based perspective. The example of these quality models are McCall's model [4], Boehm's model [5], and ISO/IEC 9126 [10-13]. The quality models assist us to systematically define abstract meaning of software quality by decomposing it into different quality aspects of different software artefacts. Quality aspects are often referred to as the quality factors or quality characteristics. Both quality aspect and artefact may be defined on different level of abstraction. They range from abstract "ilities" (e.g., maintainability) of software to concrete technical properties of certain development products (e.g., documentation level of a source code). Quality aspects may, and often are, further redefined in a hierarchical quality model into quality sub-aspects, and sub-sub-aspects until a measurable attributes can be defined. These measurable attributes are known as the quality metrics and can be used to measure different software artefacts, e.g. software requirement, software end-product or software development process [17].

Software quality assessment has two main objectives that result from different perspectives, such as, customer perspective and software manufacturer perspective. Firstly, the customers aim to assess their satisfaction toward the software product by considering its delivered functions, the behaviour of individual function against their expectation, and their business related constrains (e.g., the budget) [17]. Secondly, the software manufacturer needs the software quality assessment to justify the conformance of the software product to the predefined specifications during development and before release. The measurement of software quality has two main streams, i.e. direct quality measurement and indirect quality measurement. However, the dominant measurement in software quality is indirect quality measurement due to the abstract nature of software quality. Nevertheless, most of the indirect measurements are derived from the direct measurement. For example, the learn-ability of usability quality factor can be measured in term of the speed of users being able to operate the software independently and the measurement is derived from one direct quality measurement that collecting the total time users are trained.

Software quality assessment is usually difficult because of different preferences of various software stakeholders. For example, the online social networking system, one can probably expect that software users value more on usability aspect, the developers value more on the maintainability and the software management team value more on efficiency due to the required resources and scales of the product. Besides, the nature of the software product also gives direct influence to the stakeholders to have different preference of the quality aspects. For example, the reliability aspect is more important in financial related system and the functionality can be more important for mobile-phone application. Therefore, many approaches have been introduced in order to incorporate the different preferences from stakeholders and the product nature. The examples of them are expert judgement [3], NFR-framework [37], and Software Quality Function Deployment [36]. However, they are more software development-related and lacking a systematic way to model the preferences of stakeholders in order to assess the software quality based on the varying preferences.

Although the aforementioned approaches have attempted to aggregate the performance of different quality aspects to constitute the overall quality performance of a software product, they do not consider the possibility of various interests from the stakeholders into the assessment. Therefore, some other studies [1, 35, 38] have tried using Multi-Criteria Decision Analysis (MCDA) methods [14]. In the assessment of software quality, these studies suggested to apply MCDA method for aggregating:

1. the performance of different quality attributes and/or quality aspects
2. different priority of the quality attributes and/or quality aspects

However, their choice of MCDA method in the studies is not sufficiently justified. Most of them justify their choice of MCDA methods based on single aspect, for example, the popularity of the method [1, 35, 48, 50, 51] or the nature of aggregation function for MCDA method [49]. However, to our opinion, many other factors can derive the suitability of MCDA methods in software engineering discipline. For example, the capability of the method to support group decision making as the decision in the software quality assessment is usually conducted in a group of software stakeholders. Besides, their justification is initiated from the MCDA method perspective instead of from the software quality assessment perspective.

Therefore, we believe that it is essential to select the suitable MCDA methods and adapts to the assessment context prior to the application of MCDA method in software quality assessment. To achieve this, it is suggested to:

1. select candidate MCDA methods applicable for assessing software quality.
2. identify the weakness of the candidate methods with respect to software quality assessment.
3. improve selected candidate MCDA method regarding discovered weakness.

This thesis work covers only the first two issues. With respect to the third issues we restricted the suggested method's improvements to those suitable methods we found in the related literature.

1.2 Thesis Objective and Research Goals

The objective of this thesis is to select MCDA method(s) that is the most suitable for the purpose of software quality assessment and propose their potential adaptations based upon their remaining deficits with respect to quality assessment. To address this objective, four research goals are formulated and their details are summarized in Table 1.

Table 1: Summary of the research goals of this thesis

ID	Research Goal
G1	Identify common MCDA methods and analyze them for the purpose of identifying and understanding their basic components and characteristics.
G2	Identify the criteria for the purpose of selecting MCDA methods that are most suitable for software quality assessment.
G3	Define systematic procedure for selecting and suggesting the MCDA method(s) that is best suitable in accordance to the requirements of the software quality assessment.
G4	Select MCDA methods candidates suitable for assessing software quality and identify their weaknesses.

1.2.1 Research Questions

To address each research goal, one or more research questions are formulated. For the first research goal, G1, two research questions are formulated namely: *RQ 1.1 What are the core MCDA methods?* and *RQ 1.2: What are basic elements common for all MCDA methods?*. These two questions have to be answered in order to provide a list of core MCDA methods and the common aspects that are used to compare them. For the second research goal, G2, two research question *RQ 2.1 What are the most relevant characteristics of the MCDA methods to be considered when selecting MCDA method for general decision making problems?* and *RQ 2.2: What are the most relevant requirements regarding software quality assessment?* are formulated. These two questions provide the information about the capability of MCDA methods with respect to each method aspect and the motivation about the selection requirements in term of software quality assessment.

Research question *RQ 3.1: How to systematically select the most suitable MCDA method based on the criteria defined in RQ 2.1 and RQ 2.2?* is formulated to address the third research goal, G3, where a systematic way in selecting suitable MCDA method in software quality assessment is expected.

To propose a list of suitable candidate MCDA methods and the solution to overcome their deficits in software quality assessment, two research questions are formulated to address the last research goal G4, namely research question *RQ 4.1: Which existing MCDA methods are most suitable for software quality assessment?* and research question *RQ 4.2: What are remaining deficits of MCDA methods identified in RQ4.1 and what are potential solutions to these deficits?* .

As a summary, Figure 1 provides an overview about the relationship between the research aim, research goals and the research questions and Table 2 gives a short overview of the research questions that will be answered throughout the course of this thesis. As illustrated in Table 2, a total of seven research questions are formulated for this thesis work. The research questions RQ 1.1, RQ 1.2, RQ 2.1 and RQ 2.3 are providing some insight in the fields of the multi-criteria decision analysis (MCDA) method and software quality assessment method. The results from these four research questions help to formulate a solution in addressing research questions RQ 3.1, RQ 4.1 and RQ 4.2. In Chapter 3 we explain in more details the way we address each research question.

The research questions are formulated based on the concept of GQM paradigm [52] where each research goal is refined into one or more research questions and consecutively into the expected outcomes which will assist to supply the necessary information to answer those questions. As a result, the research questions for this thesis are categorized into two types where on one hand the answer of the research questions provide the supporting information for the research goals and on another hand the answer of the research questions (i.e. research questions RQ 2.2, RQ 3.1 and RQ 4.1) are served the main contributions of this thesis work.

Table 2: Research questions formulated for this thesis work

ID	Research Questions	Type
RQ 1.1	***What are the core MCDA methods?*** Identify different MCDA base methods that are compared or evaluated in the studies. The frequently discussed methods in association with the decision making are identified as the *core MCDA methods*.	Supporting questions
RQ 1.2	***What are basic elements common for all MCDA methods?*** For the papers selected in the primary studies, the aspects from which the components/facets of MCDA method, that are being assessed or compared, are identified and categorized in accordance to their similarity. These abstracted categories of the identified MCDA aspects are known as the common basic element of MCDA methods. Each basic element is fitted into one of the four main concept of the MCDA to understand what is actually measured.	
RQ 2.1	***What are the most relevant characteristics of the MCDA methods to be considered when selecting MCDA method for general decision making problems?*** For each of the identified basic element, its measure is extracted from the papers to get an overview of how the basic elements are actually assessed and to which extent. The relevancy of the characteristics is	

	determined based on how often the basic elements are considered in the studies.	
RQ 2.2	***What are the most relevant requirements regarding software quality assessment?*** The publications that discuss the factors that can influence the software quality assessment are first identified and then these findings are abstracted to formulate the requirements regarding the software quality assessment method.	Thesis contribution
RQ 3.1	**How to systematically select the most suitable MCDA method based on the criteria defined in RQ 2.1 and RQ 2.2?** The results from RQ 1.1, RQ1.2, RQ 2.1 and RQ 2.2 are used as the inputs for this research question to formulate and propose a selection framework to assess the suitability of MCDA method in the context of software quality assessment.	
RQ 4.1	**Which existing MCDA methods are most suitable for software quality assessment?** Reusing the previously answered information (i.e. RQ 1.1, RQ 2.1 and RQ 2.2), which MCDA methods best fit in the environment of software quality assessment are determined.	
RQ 4.2	**What are remaining deficits of MCDA methods identified in RQ4.1 and what are potential solutions to these deficits?** For each suggested MCDA methods from RQ 4.1, its discovered deficit(s) is determined and its adaptation or solution is identified from the literature.	

1.2.2 The Expected Outcomes

The contributions of this thesis are threefold:

- analysis and synthesis of two systematic literature reviews gives an overview of the state-of-art in comparing MCDA methods and requirements in software quality assessment method
- selection framework to assist quality assessors in evaluating candidate MCDA methods and in selecting the most suitable one on the basis of the information collected in both literature reviews.
- proposal of a list of suitable MCDA methods for software quality assessment, their deficits, and their possible adaptations.

To acquire the aforementioned contributions, all research questions have to be addressed. Each answer of the research questions is expected to give one or more outcomes. An overview of the expected outcomes with respect to each research question and research goal is illustrated in Table 3.

Table 3: Mapping of expected outcomes to corresponding research goals and questions

Research Goals	Research Question	Expected Outcomes
G1	**RQ 1.1**	• List of core MCDA methods.
G1	**RQ 1.2**	• Description of basic elements of multi-criteria decision making.
G2	**RQ 2.1**	• The most relevant characteristics of MCDA methods that need to be considered for selecting appropriate MCDA method in the context of general decision problems.
G2	**RQ2.2**	• Requirements on the quality assessment method.
G3	**RQ 3.1**	• Systematic procedure for selecting MCDA method most suitable for software quality assessment.
G3	**RQ 4.1**	• A list of candidate MCDA methods for software quality assessment.
G4	**RQ 4.2**	• Remaining deficits of candidate MCDA method with respect to software quality assessment • List of potential solutions for deficits of selected MCDA methods.

1.3 Structure of Thesis

This thesis report consists of nine chapters. The first chapter provides an introduction to the motivation that leads us to conduct this thesis work. Besides, the research objectives of this thesis are briefly introduced together with the expected outcomes of the thesis.

Chapter 2 illustrates a brief survey of the software quality assessment and multi-criteria decision analysis (MCDA) research fields. The fundamental concept about software quality is deliberated in Section 2.1 and then the classical approach in quality modelling is further discussed in Section 2.2. Then this chapter provides a brief introduction about software quality assessment in Section 2.3. A short introduction about the concepts and terminologies in multi-criteria decision analysis (MCDA) is discussed in Section 2.4.

Chapter 3 briefly introduces the research methodologies that are selected for this thesis work in Section 3.1. Further to that, Section 3.2 presents how each research question is addressed with the help of certain research methodology and the sequence of execution.

Chapter 4 presents the description about the systematic literature reviews in this thesis in threefold. Firstly, Section 4.1 briefly examines the need to conduct the reviews. Secondly, Section

4.2 defines the review protocols for both systematic reviews conducted in this thesis. Lastly, Section 0 briefly summarizes the immediate results collected from both reviews.

Chapter 5 deliberates the synthesis results of both systematic reviews that addressing four research questions. Section 5.1.1 describes the results collected to have a list of core MCDA methods that answering research question RQ 1.1. Section 5.1.2 presents the synthesized results to formulate the aspects of MCDA method that answering research question RQ 1.2 and Section 5.1.3 deliberates further about the characteristics for each aspect of MCDA methods that addressing research question RQ 2.1. Section 5.1.4 discusses the analysis from the second review to formulate a list of selection requirements with respect to software quality assessment method that answering research question RQ 2.2.

Chapter 6 explains in procedures that we formulate called MCDA-SQA selection framework. This framework consists of six phases. The first phase, characterization of the application domain, is presented in Section 6.1. The second phase, methods elimination, is elaborated in Section 6.2. The third phase, generalization to decision situation, is described in Section 6.3. The fourth phase, comparative analysis, is explained in Section 6.4. The fifth phase, assessment, is explained in Section 6.5. The last phase, refinement, is deliberated in Section 6.6. This chapter is ended with an example scenario to elaborate the concept of the proposed framework in more detail.

Chapter 7 explains the application of the MCDA-SQA selection framework to propose one or a list of MCDA methods that are applicable in software quality assessment. Section 7.1 explains how the result collected from first review can be utilized to formulate a list of selection requirements. Section 7.2 explains the reason the second step of the framework is skipped. Section 7.3 presents the selection criteria with respect to each selection requirement. Section 7.4 presents the comparative analysis for the core MCDA methods with respect to each selection criterion and the definition of preference for each selection criterion. Section 7.5 assesses the performance of each MCDA method with respect to each selection criterion according to its expected preference. Section 7.6 briefly provides the identified solution to adapt the MCDA methods in the context of software quality assessment and therefore give the refined assessment results.

Chapter 8 discusses the possible threats to validity identified during the course of this thesis work which are classified into four categories based on the concepts from Wohlin et. al [33]. Section 8.1 introduces the identified threat to internal validity and its consequence and solutions. Section 8.2 discusses the threats to conclusion validity in this thesis and Section 8.3 presents the threats to external validity together with their consequences and solutions. This chapter is ended with Section 8.4 that discusses the threats to construct validity and their possible consequences and solution.

Chapter 9 provides an overview of this thesis from reviewing the extent of achievement for each research goal until the concluded results for each research question in Section 9.1. Besides, Section 9.2 provides insight about some interesting direction to continue this work in the future

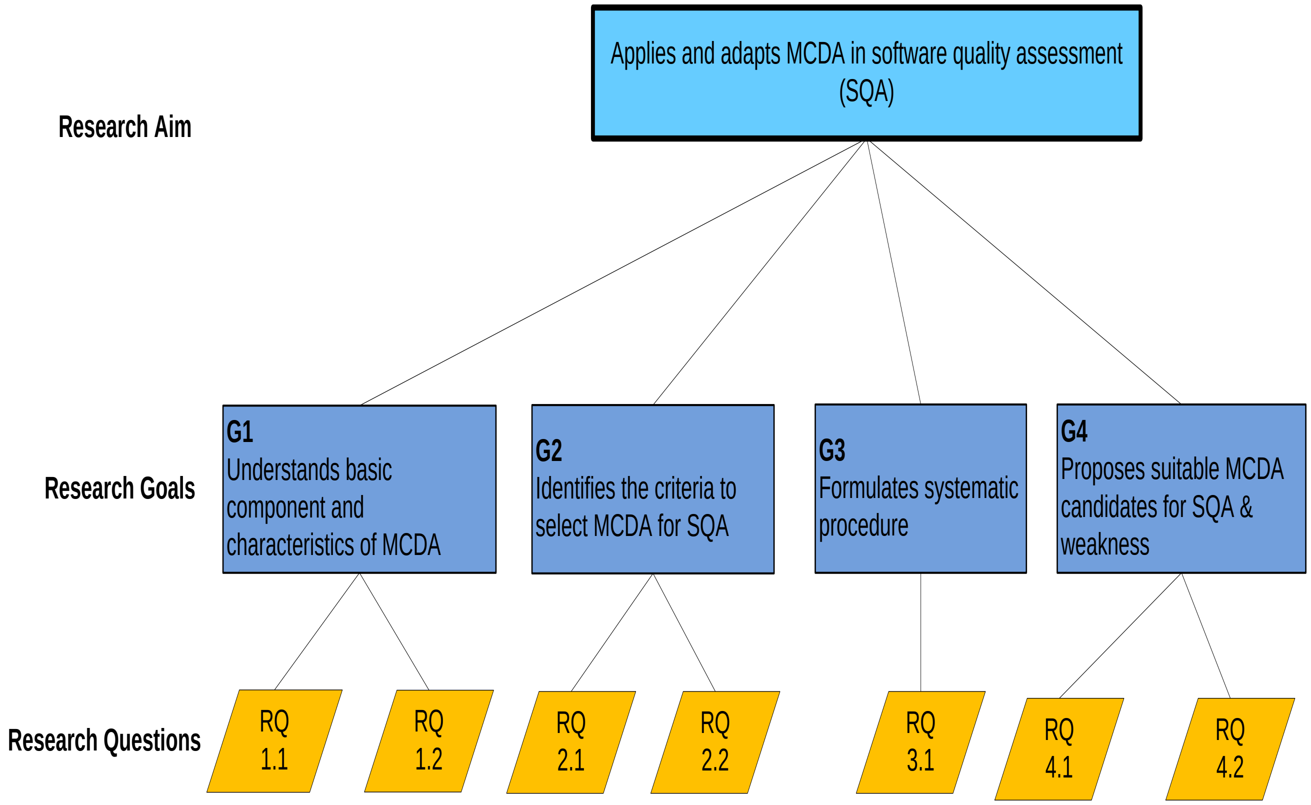

Figure 1: Relationships between research aim, research goals and the research questions.

2 BACKGROUND

This section provides background information about software quality assessment and MCDA methods. Since software quality resides in the core of software quality assessment, a brief introduction about the definition of software quality and quality modelling are also given. Section 2.1 describes the meaning of software quality and the respective concerns in defining software quality. Section 2.2 briefly introduces different quality models available and the issues of the quality models. Section 2.3 discusses the assessment of software quality in general. Finally, Section 2.4 presents the essential concepts of multi-criteria decision analysis (MCDA).

2.1 Definition of Software Quality

Fundamentally, software quality is known as the conjunction between the extent the software product able to function as expected and the satisfaction of user in using the software product [17, 21, 39]. David Garvin [16] has concluded that the "quality is complex and multifaceted" but can be defined from five perspectives, namely transcendental perspective, user perspective, manufacturing perspective, product perspective, and value-based perspective. Each of these five perspectives defines the software quality differently and the details are summarized in Table 4 [16, 17]. According to Kitchenham and Pfleeger [17], each perspective affects the way the software quality is defined and thus influences the corresponding assessment methods to be selected.

Table 4: Influences of five perspective to software quality adapted from [16, 17]

Perspective	**Definition of Software quality**
Transcendental	The software quality can be recognized but vague.
User	The software quality is viewed as the extent the software product accomplishes the users need.
Manufacturing	The software quality is viewed as the extent the software product behaves as specified in requirement.
Product	The software quality is viewed as internal characteristics of the software product.
Value-based	The software quality is viewed as the result of different priorities from different stakeholders.

Despite different views of software quality, the definition of software quality can be generalized to achieve one of these two distinct categories [16, 22]:

- *the conformance of software product to the predefined requirement specification*
  The software quality is related to a set of measurable attributes and therefore it is defined as the degree of excellent the software product comply with a prior agreed and defined specification.

- *the extent a software product satisfies the user expectation*
  In this category, the software quality is known to be independent of any measurable attributes and therefore it is defined as the fitness of the software product to achieve the expected purposes from the customers.

2.2 Software Quality Modelling

The earlier attempt in software quality modelling can be seen from Boehm's model [5] and McCall's model [4] in which are more emphasized on the product perspective. Both models provide a list of predefined quality factors of software product and each quality factor is decomposed into a list of quality criterions which are the attributes of the software product and associated with the respective measures. McCall proposes to measure the quality factors by answering yes and no questions where value 1 is given for quality criterion with answer yes and value zero for quality criterion with answer no. Each quality factors is measured as the mean value for the number of yes answer of its corresponding quality criterion. Similarly, the software quality is assessed as the mean value of number of achieved quality factors (answer yes) in the bottom-up manner. According to Kitchenham and Pfleeger [17], the approach suggested by McCall has three main problems as following:

- All quality factors shares the same priority
- Does not differentiate the variances in the degree of subjectivity between different quality factors
- Likert scale used in this approach is not expressive enough.

In the later day, ISO/IEC 9126 standard [10] has been introduced as international guidelines to standardize and generalize the definition of software quality and the way to assess software quality. The most recent version of ISO/IEC 9126 consists of four parts as illustrated in Table 5. This quality model classifies the software quality into six quality characteristics and each quality characteristic is further decomposed into a set of sub characteristics. Table 6 provides an overview of this six quality characteristics and their respective sub-characteristics in ISO 9126 [10]. Each of the sub-characteristics is further mapped to a list of attributes which are the properties of software product that can be measured by using predefined measure. All the quality characteristics are assessed based on internal measure [11] and external measure [12]. The quality-in-use measure [13] aims to assess the software product in the user environment instead of the software properties and it consists of effectiveness, productivity, safety and satisfaction.

Table 5: Purposes for different parts of ISO/IEC 9126 standard adapted from [10].

ISO 9126	Purpose
ISO/IEC 9126-1	Provides the definition of the software quality
ISO/IEC 9126-2	Provides the internal measures of software quality.
ISO/IEC 9126-3	Provides the external measures of software quality
ISO/IEC 9126-4	Provides the quality-in-use measure of software quality

Table 6: Quality characteristics and their sub-characteristic in ISO/IEC 9126 adapted from [10].

Quality Characteristic	**Sub-characteristics**
Functionality	Suitability
	Accuracy
	Interoperability
	Compliance
	Security
Reliability	Maturity
	Recoverability
	Fault Tolerance
Usability	Learn-ability
	Understand-ability
	Operability
Efficiency	Time behaviour
	Resource behaviour
Maintainability	Stability
	Analyzability
	Changeability
	Testability
Portability	Install-ability
	Replace-ability
	Adaptability
	Conformance

However, the standard quality model does not provide specific description about how to acquire the overall software quality and they only provide a list of predefined measures for individual software quality factor. Therefore, many studies have been conducted to seek a way to aggregate the results of quality factors to acquire the overall software quality. Some of the popular discussed methods are rating method [18], expert judgments [3] and rule-based classification methods [19].

2.3 Software Quality Assessment

In general, software quality can be assessed by measuring the extent of each quality aspect in a quantifiable manner. The aforementioned quality models play important roles to provide the general definition of quality aspects for a software product and give the standard measurement for each quality aspects. However, these quality models provide neither guideline nor measurement to aggregate the measurement of each quality aspects to assess the software quality as a whole [17]. Therefore, subjective rating is conducted in order to aggregate the measurement of each quality aspects [22].

Despite the measurement provided by the formal quality models, many software manufacturers also bases the quality assessment of their software product on their own defined measurements which reflects more the actual usage of their software products or suits to their testing works [25]. These measurements can be classified into three categories namely defect-based quality measurement, usability measurement and maintainability measurement [22].

In defect-based quality measurement, the software quality is equalled as the number of identified defects with respect to the specifications where the fewer defects found indicates the better software quality. Although this type of measurement provides a useful insight, the defects are discovered during the software development process and it is questionable whether the discovered defects can really lead to an operation failure [25]. So, higher defect level does not always imply lower software quality.

In the second category, the software quality is assessed by measuring the extent of its usability from the user perspective which means better software quality provides better usability or user satisfaction. Besides, the software quality assessment based on maintainability measurement is conducted by capturing the maintenance related process measures. For example, the total time needed to fix a software fault.

However, the software quality assessment methods from these three categories do not include the varying and probably conflicting priorities from different stakeholders about the quality factors involved, in particular in deciding the acceptance of the software product for their own use or launching. As a result, a number of software quality assessment related studies [1, 2, 8, 9, 25, 35] have attempted to use multi-criteria decision analysis (MCDA) method in assessing software quality.

2.4 Multi-Criteria Decision Analysis (MCDA)

Generally, a decision problem occurs when a decision maker or a group of them have a list of known alternatives on hand and need to determine which one of the known alternatives can suit their need or achieved their ultimate goal in the most optimum way [14, 23]. There are two types of the decision problem [14, 23, 24]:

- *single-criterion type* is regarded as the derivation of a decision problem from single point of view
- *multi-criteria type* is regarded as the derivation of a decision problem from multiple points of view

The points of views here are also known as a list of decision criteria or decision factors. In the real world situation, the single-criterion type of decision problem is deemed to be insufficient to support the decision making [24] and therefore multi-criteria decision problem is the dominant stream in the area of decision making.

According to Vansnick [26], a multi-criteria decision problem is structured as 3-ple {A, C, P} where:

- *A* is the potential actions or possible alternatives for a decision problem under the evaluation of MCDA.
- *C* is the set of decision criteria or decision factors which are used to assess the known alternatives.
- *P* is the performance assessment for the alternatives in order to agree on their desirability with respect to all decision criteria.

Since the nature of multi-criteria decision problem involves conflicting criteria, the decision maker finds it is essential to have a systematic approach to search the optimum solution in accordance to the related but conflicting decision criteria. Besides, a group of decision makers involved in the same decision making always possess different preference even for the same set of decision criteria and therefore their preferable decision can be different which complicates the decision making in finding the optimum solution [24, 31, 43]. Consequently, a formalized approach is needed to guide decision maker(s) in resolving the decision problem by considering multiple criteria and the existence of different preferences.

To deal with the aforementioned issues in resolving a decision problem, multi-criteria decision analysis (MCDA) or also known as multi-criteria decision making provides the formalized approaches that can help decision makers to make a better decision in selecting an optimum alternative(s) that consider the variance of preferences and the influence of multiple conflicting criteria [24].

2.4.1 The Role and Definition of the Alternatives or Potential Actions

In MCDA, the alternatives constitute the object of decision for the decision maker and the alternatives are always treated as mutually exclusive even there are some studies attempt to implement more than one alternative together [14]. There are two possibilities of the alternatives can occur in MCDA, i.e. infinite set of alternative and finite set of alternatives. Firstly, infinite set of alternatives is recognized as the possible solutions for a decision problem is unknown but its constraints are defined explicitly and therefore MCDA is responsible to assist decision maker constructing the alternative that can fulfil their objectives in the most optimum way. This type of MCDA is classified as multi-objective mathematical programming (MOMP) [14] but it is not the scope of this research work.

Secondly, the finite set of alternatives means there exists a set of known alternatives before the analysis is started but the constraints of the decision problem is not well-defined. In this context, MCDA is responsible to assist the decision maker to obtain the alternatives from the finite set of alternatives by either:

- choosing one or subset of the alternatives or
- classifying the alternatives into predefined clusters or
- ordering the alternatives from the best to worst or
- describing the alternatives

In MCDA terminology, the way to obtain the decision results is known as the *problematic*. However, the *problematic* in MCDA is always wrongly perceived as the problem or the object of decision itself but it should be the way the expected results are obtained after applying the MCDA techniques [14]. According to [14], there are four primary types of *problematic* in the area of MCDA, namely choice *problematic*, sorting *problematic*, ranking *problematic* and description *problematic*. The definition of each problematic type is summarized in Table 7.

Table 7: Definition of all *problematic* exists in MCDA [14]

Problematic	**Definition**
Choice *problematic*, α	The decision result is obtained as a single alternative or a subset of the potential alternatives.
Sorting *problematic*, β	The decision result is obtained and presented as a predefined cluster of similar alternatives.
Ranking problematic, γ	The decision result is acquired from an ordered collection of potential alternatives.
Description problematic, δ	The decision result is described without providing any suggestion or prescription.

2.4.2 The Role and Definition of the Decision Criteria

To resolve the decision problem, the decision maker has to first construct the set of criteria and ensure the following properties are fulfilled [14].

- Every decision maker comprehends the meaning of criteria sufficiently.
- The evaluation of criteria for each alternative has to be done without considering its relative importance which can be varied for different decision makers involved.
- The consistencies of all criteria have to be assured.

In MCDA, the criterion acts as a tool to assess and compare the desirability of an alternative based on the performance of one alternative with respect to this criterion [14, 27]. There are two possible scales to assess the performance of an alternative concerning one criterion, namely quantitative scales and qualitative scales [14]. In quantitative scale, the performance of an alternative is represented in numerical scales in which difference in two scores of the scale can be defined clearly and carries certain meaning. For qualitative scales, the performance for an alternative with respect to a criterion is represented in an order where the difference between two different orders of scale cannot be defined in an exact manner [14].

According to Vincke [23], the measure of decision criteria with respect to each alternative can be classified into four types [23] namely measurable criterion, ordinal criterion, probabilistic criterion and fuzzy criterion. The definition of each criterion type is summarized in Table 8.

Table 8: Definition of four criterion types [23]

Type of Criterion	Definition
Measurable criterion	The measure of this criterion allows the preferential evaluation of intervals of the measure scale. This type of criterion can be further classified into three sub-categories: • *true-criterion* The measure of criterion does not consider any predefined threshold. • *semi-criterion* The measure of criterion includes the predefined indifference threshold. • *pseudo-criterion* The measure of criterion includes the predefined indifference and preferential thresholds.
Ordinal criterion	This category is also known as qualitative criterion and is assessed in qualitative measure scale. The measure of this criterion characterizes only the order on the set of alternatives involve.

Probabilistic criterion	The performance of criterion is uncertain and estimated based on a probabilistic distribution.
Fuzzy criterion	The interval of criterion measure scale derives the performance of alternatives.

2.4.3 The Role and Definition of the Preference Modelling

The preference modelling in MCDA is a mechanism to determine whether an ordering relation or indifference relation exists between two objects under evaluation [14]. The most common way of preference modelling is basing on the ordering relation between two objects and help to address choice *problematic* or ranking *problematic*. The preference model based on indifference relation is responsible to study the similarity between two objects and group all objects with same features together into a predefined cluster. This type of preference model is able to address the sorting *problematic*. Guitouni and Martel [40] introduced five different preferential relations available in the area of MCDA in order to express the preference of decision maker and the details are summarized in Table 9.

Table 9: Definition of five preference relations [40]

Preference Relationship	Notation	Definition
Strict preference, P	*a* ***P*** *b*	This relation is applicable to the situation where there is sufficient evidence to conclude that alternative *a* is more preferred to alternative b. Therefore, alternative *a* is strictly preferred as compared with alternative *b*.
Weak preference, Q	*a* ***Q*** *b*	This relation is applicable to the situation where there is uncertainty between indifference situation and strictly preference situation. Alternatively, this relation is applicable whenever the strict preference is not certain.
Indifference, I	*a* ***I*** *b*	This relation is applicable to the situation where alternative *a* has no difference against alternative *b* or their differences are too small to distinct them.
Incomparability, R	*a* ***R*** *b*	This relation is applicable to the situation where hesitation occurs in deciding whether alternative a is preferred to alternative b or alternative b is preferred to alternative a. This situation occurs when a is better than b in certain set of criteria but worse than b in another set of criteria and all these criteria are not comparable.

Outranking relation	*a* ***S*** *b*	This relation is the union of the strict preference, weak preference and indifference relation. It is applicable when there is strong evidence to believe that with regard to all criteria involve an alternative *a* is at least as good as alternative b and there is no reason to oppose this conclusion.

In MCDA, it is common to numerically analyze the performance of each alternative with respect to each criterion. It is conducted in two phases namely the Scoring phase and the Weighting phase [14, 45].

- *Scoring Phase* - For each alternative with respect to criterion, a numerical score is given based on their expected consequences [45]. The consequence here is regarded as the measurement of the alternative with respect to a criterion. Usually, a more preferred alternative is given higher score on the preference scale and less preferred alternative is given lower score.
- *Weighting Stage* - Based on the interest of the stakeholders or decision maker, a numerical relative importance can be used to indicate their interest for each criterion [45]. This numerical relative importance is regarded as *weight*. A more preferred criterion is given higher numerical relative importance and the less preferred criterion is given lower numerical relative importance.

The ultimate preference of an alternative is evaluated by considering its performances with respects to all criteria. There are three distinct operational approaches to aggregate the performance of multiple criteria for an alternative: 1. building a unique synthesized criterion; 2. outranking relation and; 3.interactive judgement. Their details are summarized in Table 10.

To provide a better illustration of the concept of preference evaluation, an example is given here. Mr. Lim intends to buy a computer by considering the colour and the price. He has two candidates namely computer A and computer B. He prefers black colour and prefers the price to be lower than €500. Besides, he concerns more to the price of the computer. The actual colour and actual price of the computer designate the consequences of price criterion and colour criterion. Five points scale is decided to indicate the preference of Mr Lim based on his expectations or preference. Mr. Lim rates 3 points to indicate his interest in having black colour computer and 2 points for computer with other colours. As for the price of computer, Mr. Lim rates 3 points for the computer with prices lower than €500 and 2 points for the computer with price more than €500. Eventually, Mr. Lim rates 3 point to indicate his preference to the price of the computer.

Table 10 : Summary about three main aggregation approaches [14].

Operational approaches	Description
Building a unique synthesized criterion	The preference of its entire criterion is aggregated into a single, unique utility value and this is used to justify the desirability of the alternative. However, this approach does not allow incomparable criteria being considered in the same evaluation.
Outranking relation	The preference is modelled among alternatives based on an outranking relation that represents the preferences of decision makers.
Interactive judgment	Trade-off computation and communication about the preferences of decision makers are iterated in the trial-and-error context.

Table 11: Summary of the example

Computer	Colour		Price		Weighted-sum
	Consequence	Score	Consequence	Score	
A	Black	3	€502	2	(3x2) + (2x3) = 12
B	Blue	2	€490	3	(2x2) + (3x3) = 13

In this example, the *decision criteria* are the colour and price of the computer. The *alternatives* here are computer A and computer B. The actual value of each criterion for each computer is referred as the *consequence*.

Table 11 illustrates the meaning of consequence and score for this example. The *weight* of the decision criteria is referred to the preference of Mr. Lim between colour criterion and price criterion. Based on the given information, the weight of colour criterion is 2 points and the weight of the price criterion is 3 points.

Let the aggregation approach in this example to be weighted-sum approach where all the preferential information is summed up. Therefore, the computer A has total 12 points and the computer B has total 13 points. Eventually, computer B is suggested to Mr. Lim as the best option in accordance to his expectation.

3 RESEARCH METHODOLOGY

To address the research questions mentioned in Section 0, two research methods are utilized, namely systematic literature review [6] and conceptual analysis. Apart from that, a traditional literature review is conducted to obtain a deeper understanding of the topics under investigation, namely software quality assessment and Multi-Criteria Decision Making (MCDA). Prior to the discussion how each research methods is applied in this thesis work, the definition and general concept of each research method is illustrated in Section 3.1. Section 3.2 illustrates how the selected research methods are applied in this thesis to address the research questions.

3.1 Brief Description of Research Methodologies

Two research methods are applied in order to complete this thesis work, namely systematic literature review and conceptual analysis. Each method is briefly described in the following sections where Section 3.1.1 discusses the concept of systematic literature review (SLR) and Section 3.1.2 illustrates the meaning of conceptual analysis.

3.1.1 Systematic Literature Review

To provide unbiased and repeatable research methodology, Kichenham et al. [6] introduces a list of systematic guidelines to conduct the systematic literature review (SLR) which also known as systematic review. SLR is noted as a formation of secondary study based on the individual studies collected and contributed to the review. These individual studies are called primary studies [6]. As suggested by the guidelines of Kitchenham et al. [6], SLR is conducted in three primary phases namely, planning, conducting and reporting the review. In the planning phase, the research questions and review protocol are defined. The review protocol is constituted of the objective and the definition of searching procedures that are followed during the course of systematic review. The second phase is conducting review that involves [6]:

- searching all related primary studies based on the predefined search strategy
- extracting the interested data from the selected primary studies
- evaluating the quality of the selected primary studies
- synthesizing the extracted data.

The third phase of systematic literature review is the reporting phase where all the results collected from the review are documented.

3.1.2 Conceptual Analysis

The conceptual analysis method [42] is applied on the data gathered from systematic literature review to break down the primary studies into the constituent parts in order to acquire deeper understanding about the concerned issues with respect to each research question and analyzes each constituent to formulate the answer to address each research question of this thesis work.

3.1.3 Literature Review

Literature review is a method to study the existing knowledge on a specific subject area. There is no fixed procedure to conduct a literature review as it is varied according to the purpose and scope of the research. Nevertheless, it is essential to conduct the literature in an effective way. As suggested by Levy and Ellies, the literature review process consists of "sequential steps to collect, know, comprehend, apply, analyze, synthesize, and evaluate quality literature in order to provide a firm foundation to a topic and research method" [41].

3.2 Procedure to Apply the Determined Research Methodologies

As discussed in Section 0, seven research questions are formed in order to achieve four research goals. In this thesis work, there are two systematic literature reviews are conducted.

The first systematic literature review is conducted to answer three research questions:

- *RQ 1.1: What are the core MCDA methods?*
- *RQ 1.2: What are basic elements common for all MCDA methods?*
- *RQ 2.1: What are the most relevant characteristics of the MCDA methods to be considered when selecting MCDA method for general decision making problems?*

This systematic literature review focuses on three main elements which are considered as crucial for multi-criteria decision analysis (MCDA) methods. These are: 1. base MCDA methods, 2. aspects of MCDA method used for the evaluations of MCDA methods, and 3. characteristics of MCDA methods considering in the assessment. All the steps of the systematic review are driven by the consideration of these three aspects, which will be further discussed in Chapter 4.

The second systematic literature review is conducted to address research questions RQ 2.2: *What are the most relevant requirements regarding software quality assessment?* This review focuses on (1) the objectives of introducing assessment methods, (2) the weaknesses of the existing assessment method methods, and (3) constraints of the software engineering environment that potentially influence applicability of particular quality assessment method (e.g. incomplete data available for quality evaluation). These results from these three aspects formulate a list of requirements to be used later in justifying the suitability of MCDA methods in SQA. The results collected from this review are analyzed and consolidated with the results from an existing industrial survey [47].

Conceptual analysis is the research method applied in this thesis work to address two research questions:

- *RQ 3.1: How to systematically select the most suitable MCDA method based on the criteria defined in RQ 2.1 and RQ 2.2?*
- *RQ 4.1: Which existing MCDA methods are most suitable for software quality assessment?*

The results collected from both systematic reviews serve as the main inputs for the conceptual analysis. Six steps of conceptual analysis [42] are incorporated in the data extraction strategy (refer to Section 4.2.5) of both systematic reviews to systematically gather data for later use in the conceptual analysis. These steps are [42]:

- determining the analysis level
- determining the number of concepts to code for
- determining the existence of code for a concept
- determining the way to differentiate the concepts
- formulating the rules to code the text
- determining the way to handle "unrelated" information

The output of these six steps is reflected as the data extraction form (refer to Table 16 in Section 4.2.5) for both reviews. This has illustrated the classification and generalization done in the reviews. The seventh step of conceptual analysis (refer to step "coding the text" in [42]) is completed by constructing the templates for the data extraction forms use in both systematic literature reviews (refer to Table 18 in Section 0 and Table 19 in Section 4.3.5). The data gathered from the systematic literature review and the respective results (refer to Chapter 5) lead to the eighth step of conceptual analysis (refer to step "analyze the result" in [42]). With the findings from the conceptual analysis, the essential concepts in selecting MCDA methods for software quality assessment are proposed to formulate a selection model which addresses the research question RQ 3.1.

Apart from that, the finding from the conceptual analysis is also used to instantiate the proposed selection model in order to suggest the suitable MCDA methods for software quality assessment. Based on the suggested MCDA methods, literature review is conducted to seek possible solution to overcome the weakness(s) of the methods in software quality assessment and this result addresses the research questions "*RQ 4.2: What is remaining deficits of MCDA methods identified in RQ4.1 and what are potential solutions to these deficits?*".

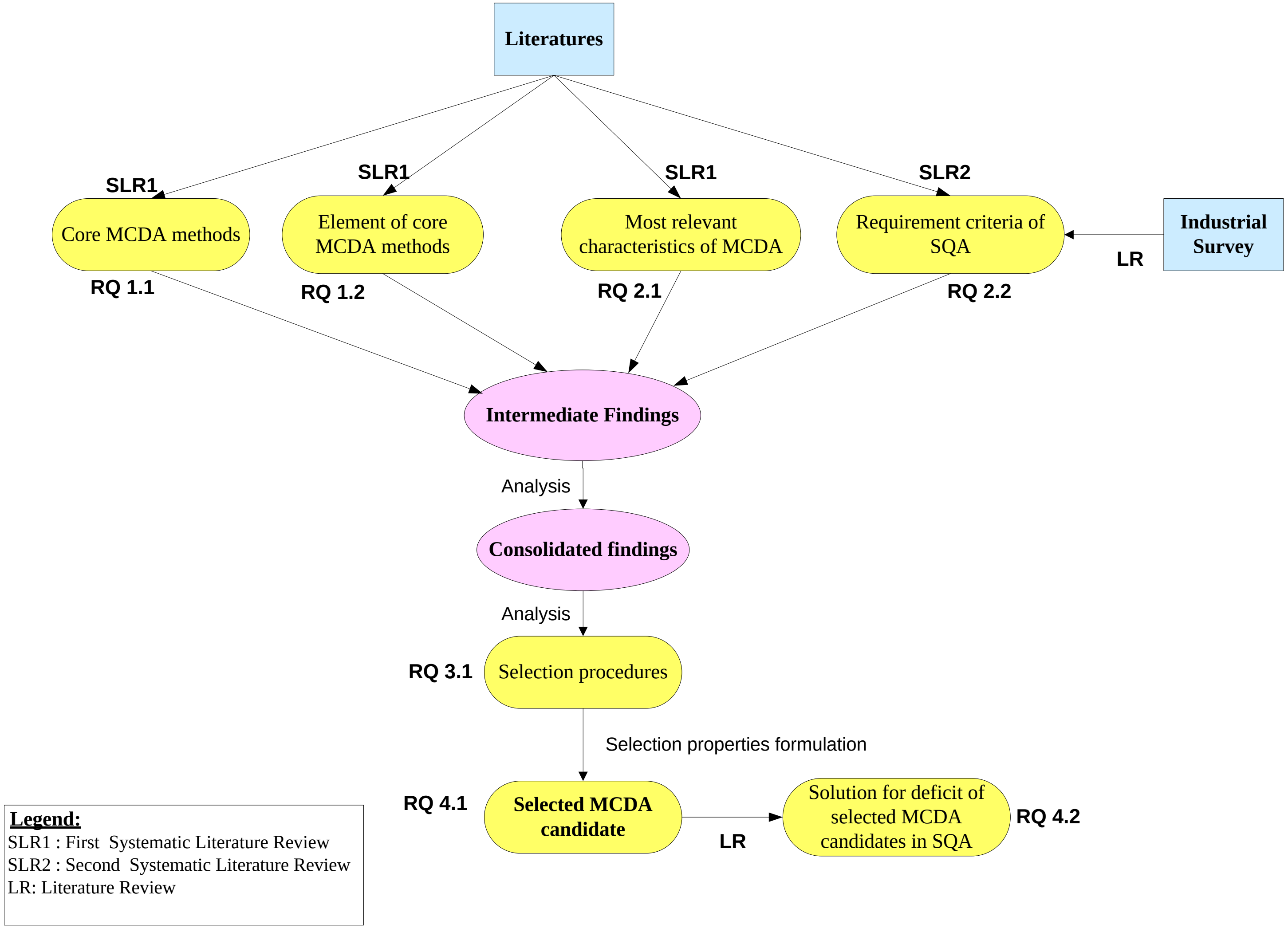

Figure 2: Flow chart of the research methodologies in this thesis work

4 SYSTEMATIC LITERATURE REVIEW DESIGN AND EXECUTION

In this thesis, the systematic literature review procedures suggested by Kitchenham et al. [6] are followed. It consists of three main phases namely "planning the review", "conducting the review" and "reporting the review" [6]. This chapter presents the way the first two phases are conducted in this thesis. The last phase "reporting the review" is documented in Chapter 5.

Section 4.1 deliberates the motivation to conduct the systematic literature reviews for this thesis. Section 4.2 defines the review protocols to be followed in the systematic literature reviews. Section 0 summarizes the immediate findings gathered from conducting both reviews.

4.1 Need of Systematic Review

To determine whether it is necessary to conduct a systematic review on the thesis topic, a preliminary search is performed and the search shall uncover the following:

- *Was systematic review regarding applicable MCDA methods in software quality measurement already executed and to which extent?*

Compendex[1] and Inspec[2] databases have been selected in this preliminary search because of their large volume of literature they comprise.

4.1.1 Search String Used in Preliminary Search

("MCDA" OR "multi-criteria decision analysis" OR "multi-criteria decision management" OR "MCDM") AND ((software and quality) AND (evaluat* OR measur* OR assess* OR estimat*))

Based on the search string above, a preliminary search is conducted in Inspec and Compendex Digital libraries by using Engineering Village[3] to discover the literature review on selecting and applying MCDA methods for SQA. The result obtained is 16 papers with 2 duplicate papers. By reading their title, 5 papers were found not related in the field of software engineering or computer science.

After reading the abstract, the remaining publications are found irrelevant because they focus on applying single MCDA methods in software quality evaluation without justifying their selection of MCDA or validating their selection with the other MCDA methods which can be applicable in software quality evaluation. The preliminary search result has merited the author to conduct two systematic literature reviews in the area of multi-criteria decision analysis (MCDA) methods and software quality assessment method respectively.

[1] http://www.engineeringvillage2.org/controller/servlet/Controller
[2] http://www.iee.org/Publish/INSPEC/
[3] http://www.engineeringvillage2.org/controller/servlet/Controller

4.2 Review Protocol

A review protocol is a crucial aspect in systematic literature review because of its documentation of the detailed steps to be carried out in order to mitigate the chance of the bias from the researcher [6]. This section presents the review protocol and the corresponding steps that are planned to implement both systematic literature reviews.

4.2.1 Search Strategies

The search strategies are formulated to decide the primary studies that are to be selected for this thesis. In this thesis, two distinct searches have to be performed in the area of:

- multi-criteria decision analysis (MCDA)
- software quality assessment (SQA)

For these two searches, the search strategies consists of the data sources selected for the searches, the corresponding search strings and the general rules recommended to be applied in the searches.

Data Sources

Based on the suggestion given by Kichenham et al [6], five data sources are selected from seven suggested data sources to perform the search in this review. Compendex and Inspec are selected as the first two reference data sources in the search because of its large volume of publications, in particular the engineering and computer science related publications.

Besides, these two data sources can be accessed using Engineering Village[4] interface that can ease the review due to its advance search capability and user friendliness. SCOPUS[5] is also selected due to the same reason as the previous two databases. In order to reduce the risk of missing important papers, IEEE Xplore[6] and ACM Digital database[7] are also included in this search because of their popularity in the area of engineering and computer science.

The second review is separately conducted in order to assess the state of practice of software quality assessment in software engineering. In aforementioned five digital libraries, the second review focuses on the publications from the proceedings of five leading software engineering conferences:

- International Conference on Software Engineering (ICSE)
- International Symposium on Empirical Software Engineering and Measurement (ESEM)
- International Conference on Software Engineering and Knowledge Engineering (SEKE)
- Product Focused Software Development and Process Improvement (Profes)
- Software & Systems Engineering Essentials (SEE)

---

[4] http://www.engineeringvillage2.org/controller/servlet/Controller
[5] http://www.scopus.com/home.url
[6] http://ieeexplore.ieee.org/Xplore/guesthome.jsp
[7] http://portal.acm.org/portal.cfm

We decided to focus on conference proceeding as they are typically the first published source of the latest results from research and practice (journal publications typically involves certain delay, e.g., due to review process). The choice of the conferences is based on our personal perspective and subjective evaluation of their importance. Yet, we could not find any reliable source that ranks software engineering conferences. The scope restriction we took in the second review can cause a threat to the external validity of this thesis work which we further discuss among the other threats in Chapter 8.

Search String

Two sets of search strings are formulated and illustrated in Table 12 and Table 13, respectively for each search or systematic literature review (SLR). Both search strings are resulted from a few discussions and have gone through a few modification sessions. Furthermore, both search string are applied to Title/Abstract/Topic/Subject search fields which depends on individual data sources that is used.

During the background studies, it is known that the multi-criteria decision analysis concept has been popular for a few decades and therefore a vast number of publications can be expected if the search string includes only the MCDA keywords. Besides, this review focuses on how the behaviours of the MCDA methods are evaluated. By considering these two reasons, the search string for first review consists of three keywords: multi-criteria decision analysis (MCDA), method and evaluation. The MCDA keywords consist of four possible synonyms: multi-criteria decision analysis, multi-criteria decision making, MCDA (i.e. the abbreviation) and MCDM (the abbreviation). The method keywords are method, approach and technique. The evaluation keywords are evaluate, assess, select, compare, review and overview. After experimenting with several possible combination and variations of three types of keywords, the search string is determined as Table 12 where the character “*” indicates the wild character to represent all possibilities of the root word for the keywords. For example, assess* means different variation from root word assess which can be assessing, assessment, assesses, and so on.

Table 12: Search strings formulated for SLR in MCDA method

MCDA keywords	Op	Method keywords	Op	Evaluation Keywords
("MCDA" OR "multi-criteria decision analysis" OR "multi-criteria decision management" OR "MCDM")	AND	(method* OR approach* OR technique*)	AND	(classif* OR evaluate* OR assess* OR select* OR compar* OR overview OR review*)

Op: Operators

It is important to note that there exists other possible synonyms for the keywords but the search strings are determined to be as aforementioned. However, the prior experiments that include more possible keywords have returned a vast number of publications (some even more than 5,000 papers) which is unfeasible to finish within the allowed timeframe. It is our intention to have the publications in the range of 700 to 600 papers from each data source in order to finish this thesis within the given time-frame.

The second review focuses on the software quality assessment and therefore its search string should consist of two types of keywords: assess and software product quality. During the background study, it is known that the software quality and software product quality are sometime used interchangeable in the literatures. Therefore, the keywords for software product quality is consisted both of them. The keywords of assessment are the union of all synonyms suggested in the literature of the background studies and also with the help of English dictionaries. The assessments keywords are assess, evaluate, measure, review and estimate. After a few experiment on different combinations of the keywords, the final combination of the keywords for this review is illustrated in Table 13 in order to have reasonable number of publications returned from each data sources which is less than 200 papers.

Table 13: Search strings formulated for SLR in SQA

Software product quality keywords	Operator	Assessment Keywords
(“software quality” or “software product quality”)	AND	(assess* OR evaluat* OR measur* OR review* OR estimate*)

4.2.2 Study Selection Procedure

Two iterations of the selection procedure are executed where the first iteration focuses on MCDA methods search pool and the second iteration aims for the software quality assessment. The following steps are carried out during each of the iterations.

1. The search strings are applied to retrieve papers in each of the data sources.
2. The relevancies of the retrieved papers are checked based on the selection criteria in Section 4.2.3 with respect to the search pool involved. The non-relevant papers are removed from the search pool.
3. If the decision is uncertain, the conclusions of the papers have to be checked for assurance.
4. All selected papers from five data sources are consolidated and the duplicate papers are removed.

4.2.3 Study Selection Criteria

In general, the title and abstract of the papers that are retrieved from all two searches will be checked for relevancy according to the criteria summarized in Table 14. If the paper is still unjustifiable whether it is relevant, then the conclusion section of the paper will be read for further confirmation.

Besides, all papers with non-English language are excluded. The publication years for both SLR are restricted to be from year 1999 onward until year 2010.

Table 14: Selection criteria for each search pool with respect to answer the related RQs.

Search Area	Research Question	Inclusion Criteria	Exclusion criteria
MCDA	RQ1.1 RQ 1.2 RQ 2.1	• The original paper that introduce the core MCDA method. • The papers illustrate the strengths, weaknesses, classification or comparison for the individual MCDA techniques. • The paper discusses the reason for certain MCDA methods to be selected for decision making. • The papers deliberate the application of a MCDA technique in software quality evaluation or the decision making in software engineering context.	• The papers that attempt to select the combined MCDA methods in general decision making without elaborating the weaknesses of single MCDA methods. • The paper focuses on detail implementation of MCDA methods in certain context. • The implementation of MCDA tool. • The papers focus on introducing MCDA concepts instead of techniques themselves. • The application of non-core MCDA methods and hybrid MCDA methods in all domains except software engineering. • The papers uncover the deficit of individual MCDA methods and introduce the combined MCDA methods to overcome the deficit.
SQA	RQ 2.2	• The paper focuses on introducing a new software quality assessment with the reason given or identified deficit of existing software quality assessment methods. • The papers that illustrate the application of MCDA method for the purpose of the software quality assessment (SQA). • The papers that discuss the nature of software quality assessment and also the	• The paper focuses on the quality assurance related activities or processes. (e.g. inspection, testing, etc) • The paper focuses on improving the software quality.

		environmental factors affecting the assessment of software quality.	

4.2.4 Study Quality Assessment

Different from the one suggested from Kichenham et al [6], the study quality assessment in this thesis is considered as part of the inclusion criteria. This means the selected papers will have shown certain type of evidence or evaluation related to software quality assessment method and MCDA method. The term evidence includes from a simple application of the study on a toy example, to the experiment, and any other types of empirical evidence.

4.2.5 Data Extraction Strategy

After the relevant papers are selected for actual full text reading, the data extraction is performed during the full text reading. To avoid rework and reduce mistake in extracting important data from the paper, the extraction forms are compiled as shown in following to extract data answering RQ 1.1, RQ 1.2 and RQ 2.1.

Table 15: Paper meta-data

Paper Item	**Value**
Title:	
Author:	
Publication Year:	
Data source:	

Table 16: Data extraction form for MCDA methods

Group	**Data Item**	**Data Value**
Problem	Type	• Choice / Problematic α • Sorting / Problematic β • Ranking / Problematic γ • Description / Problematic δ
	Size	• Small • Medium • Large
Alternative	Number of Alternatives	• Small • Medium • Large
	Ability to adapt new alternatives	• Yes The measure of the new alternative can be included into the existing

		computation result without requiring a brand new re-computation. • No The existing computation result cannot be reused and a brand new computation is expected in order to include a new alternative.
	Handling conflicts and incompatibility	• Yes • No
	Hierarchical structure supported	• Yes • No
	Nature of alternative set	• Continuous • Discrete
Criteria	Data Type supported	• Qualitative • Quantitative • Mixture
	Measurement scale	• Measurable • Ordinal • Probabilistic • Fuzzy
	Criteria weighting	• Yes • No
	Interdependency between criteria	• Supported • Not supported
Usage	Tool available	• Yes • No
	Implementation Cost	• High It is expensive to implement the MCDA method in the software and require extra training for the evaluators to operate. • Medium It is either expensive to implement the MCDA method in the software or the related MCDA-software require extra training for the evaluator to be able to operate it. • Low It is cheap to implement the MCDA method in the software and it does not require extra training for the evaluator to operate.
	Ease of computation	• Yes

	The rapidity for the user to operate the method (without the help from any software) and the complexity of the method.	The operation of the method is simple and rapid. • No The operation of the method is complex and slow.
	Partial Evaluation	• Yes • No
	Support group decision	• Support • Not support

4.3 Systematic Literature Review Execution

Two reviews are conducted for this thesis based on the review protocol discussed in the afore-section. For each review, the respective selection results are presented and then followed by the data extracted from the selected papers.

4.3.1 Extraction from Data sources

The next step in the review is to extract the papers from the corresponding data sources based on the search strings as discussed in the aforementioned section. The details of papers extracted from each data source are tabulated in Table 17.

Table 17: Statistics result of papers retrieved from each data sources with respect to related study

Databases	Primary Study (MCDA)	Primary Study (SQA)
Compendex	630	32
Inspec	423	91
SCOPUS	859	173
IEEE	143	93
ACM	94	50
Total:	2149	439

4.3.2 Selection of Papers for Primary Studies in MCDA

From a total of 2149 papers retrieved from all digital libraries, a total of 1977 papers are found not relevant after reading publication titles and abstracts in accordance to the predefined inclusion/exclusion criteria (refer to Section 4.2.3). The duplicate papers were diminished, which resulted in reduction of the number of papers down to 118. After reading the conclusion of the publications, 42 papers are discarded as these publications focus on discussing the way to resolve deficits of MCDA method. The remaining papers are furthered classified into two categories namely MCDA comparative and MCDA methods relate with software engineering. This is because the papers in first category will be used in the primary study in MCDA (first SLR) and the papers in latter category act as supplementary data sources to primary study in SQA (second SLR).

59 papers are found related in the MCDA comparative and 17 papers are found related to the application of MCDA methods in the area of software engineering. In the final pool of papers for primary study in MCDA, 39 papers are remained after discarding non-full text papers and filtering out the papers that are found irrelevant after reading full-text. Figure 3 provides the overview of the selection of publications for first SLR.

4.3.3 Data Extraction for Primary Studies in MCDA

According to the data fields designed in Table 16, a data extraction template for MCDA pool is created using spreadsheet application as shown in Table 18. There are 39 identified primary studies are extracted based on this template and their details are summarized in Table 48 (refer to Section 11.2).

Table 18: Template of data extraction form in spreadsheet used in primary study of MCDA

Title			
Publication Year			
Research Method			
MCDA Method Type			
MCDA Base Methods			
Characteristics of MCDA methods	**MCDA Method**	**Element**	**Characteristics**
Decision Facet	**Deciding element**	**Reason/ Impact**	
Limitation & Suggestion	**Limitation**	**Suggestion**	
Focus of Paper			
Remark			

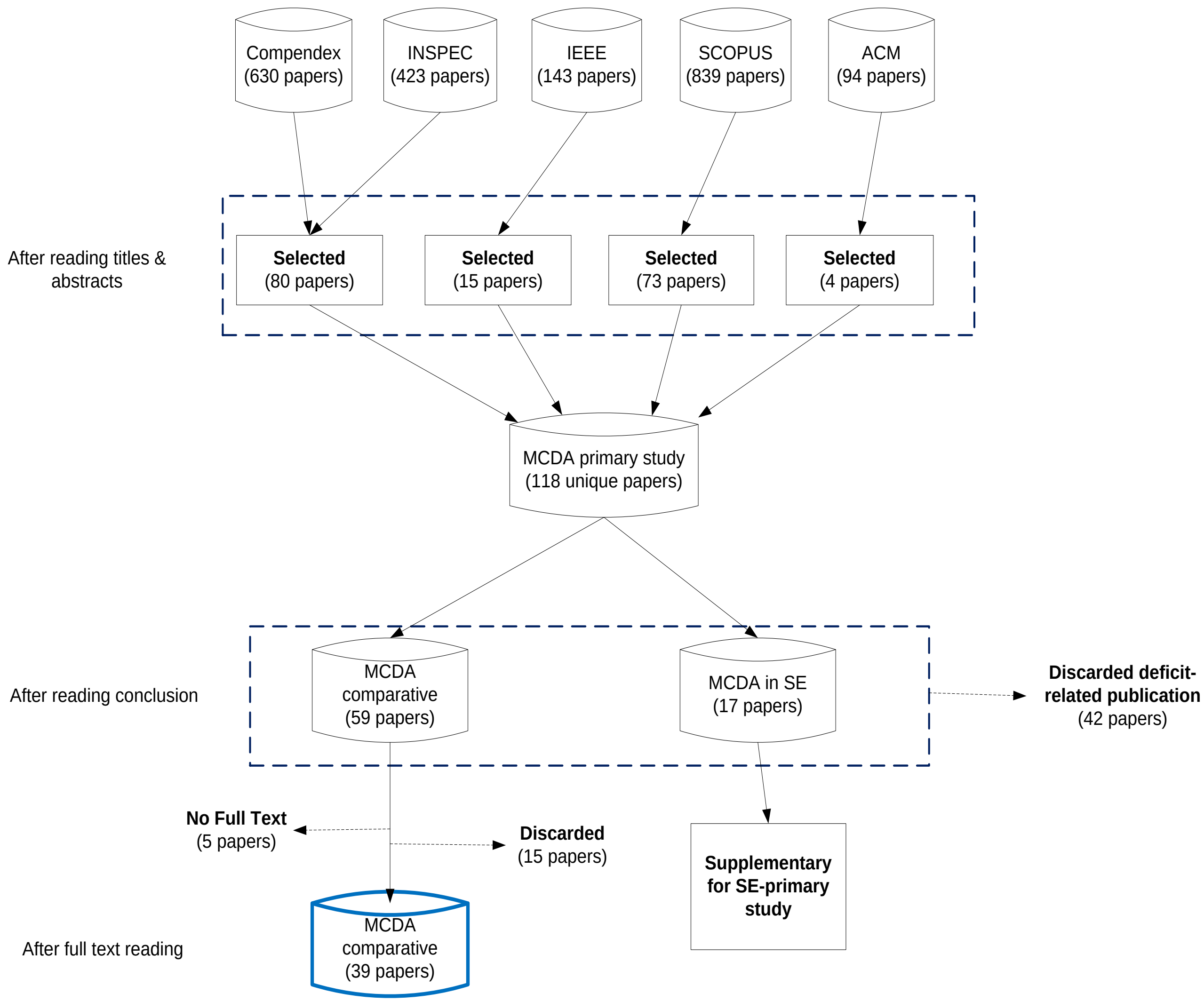

Figure 3: Overview of primary studies in MCDA selection.

4.3.4 Selection of Papers for Primary Studies in SQA

A total of 439 papers are firstly retrieved and the pool of paper is diminished to 40 papers after evaluating the title and abstracts of the papers based on the inclusion/exclusion criteria. Many papers are found related with the quality assurance activities or process instead of evaluating the software product itself and consequently a plethora of them are removed from the selection. Then, all duplicate papers are removed and this constitutes the paper pool to 28 unique papers. This paper pool is then combined with the supplementary papers from the first review and eventually the immediate pool of papers to be 42 papers. After reading the conclusion, seven papers are found not relevant and being discarded which make the final pools of paper to be 35 papers. The overview of the papers selection for the second SLR is illustrated in Figure 4.

.

Figure 4: Overview about the selection of papers related with primary study of SQA.

4.3.5 Data Extraction for Primary Studies in SQA

A data extraction template for second review in SQA is created by using the spreadsheet application and the details are illustrated in Table 19. A total of thirty-five identified primary studies in this second review are extracted based on this template. They are summarized in term of publication title and the authors in Table 49 (refer to Section 11.3).

Table 19: Template of data extraction form in spreadsheet used in primary study of SQA

<table>
<tr><td>Title</td><td colspan="2"></td></tr>
<tr><td>Publication Year</td><td colspan="2"></td></tr>
<tr><td>Research Method</td><td colspan="2"></td></tr>
<tr><td>Focus of the study</td><td colspan="2"></td></tr>
<tr><td>New SQA method</td><td colspan="2"></td></tr>
<tr><td>Reason to introduce/select the SQA method</td><td colspan="2"></td></tr>
<tr><td rowspan="2">Weakness/ strength of the existing SQA method</td><td>Weakness/Strength</td><td>Impact</td></tr>
<tr><td></td><td></td></tr>
<tr><td>Weakness/ strength of the new SQA method</td><td colspan="2"></td></tr>
<tr><td>Environmental factor</td><td colspan="2"></td></tr>
</table>

5 SYSTEMATIC REVIEW RESULT

5.1 Research Question

Four research questions (i.e. RQ 1.1, RQ 1.2, RQ 2.1 and RQ 2.2) will be addressed in the following sections (Section 5.1.1 to Section 5.1.4). Research questions RQ 1.1, RQ 1.2 and RQ 2.1 are addressed in the first review concerning the study in multi-criteria decision analysis (MCDA) methods and research question RQ 2.2 is addressed in the second review concerning the area of software quality evaluation (SQA). Each research question constitutes two sections, namely:

- Results & Analysis section provides an examination based on the answer of the research question,
- Discussion section provides further analysis and discussion based on the answer of the research question

5.1.1 RQ1.1: What are the core MCDA methods?

Result & Analysis:

To answer this research questions, all MCDA bases methods are identified from the selected publications of the first reivew and an investigation is conducted to select the core MCDA methods which are the MCDA based methods and are commonly assessed in the publications. MCDA base methods are the original evaluation methods that applying the concept of multi-criteria decision analysis to support the decision making process. For example, Analytic Hierarchy Process (AHP) method, Weighted Sum Method (WSM) method. The non-MCDA base methods are referred to the variance version of the MCDA base methods and/or the hybrid MCDA methods. These non-MCDA base methods are not the focus of this thesis and therefore a plethora of publications that assess these non-MCDA base methods have been diminished. As an example of non-MCDA base method is Fuzzy AHP [34].

A total of twenty-one MCDA bases methods used in the general decision making are identified. Figure 5 depicts the distribution of different MCDA bases methods found in the publications. From Figure 5, it is evident that Analytic Hierarchy Process (AHP) method is the mostly evaluated method in the area of decision making. ELECTRE family methods have been reported 15 times, TOPSIS method has been reported 14 times, PROMETHEE family methods have been reported for 12 times and WSM is reported for 9 times.

To determine whether the identified MCDA base method is the core MCDA method, we evaluate based on its high occurrence in the review. Twenty-one MCDA base methods are found in the review and they are treated equally important in this review. The high occurrence here is known as the number of publications the identified method possessed has achieved at least equal to the average of percentage shared among them, i.e. approximately 4.76% or 2 papers. Based on this threshold value, Interactive method, Cooperative Game Theory (CGT), Measuring

Attractiveness by a Categorical Based Evaluation Technique (MACBETH), Regime, ORESTE, Multi-Attribute Analysis (MAA), Simple Average Weight (SAW), Weighted Displaced ideal with parameter 2 (WD2), Weighted Displaced ideal with parameter ∞ (WD∞), and Multi-criteria Ranking Ordering (MRO) are classified as low occurrence due to its single occurrence in the review. Besides, their applicability is analyzed under a single domain and therefore its adaptability to another domain, i.e. software quality evaluation, to our belief, can be relatively low. Therefore, these nine MCDA base methods are not classified as core MCDA method.

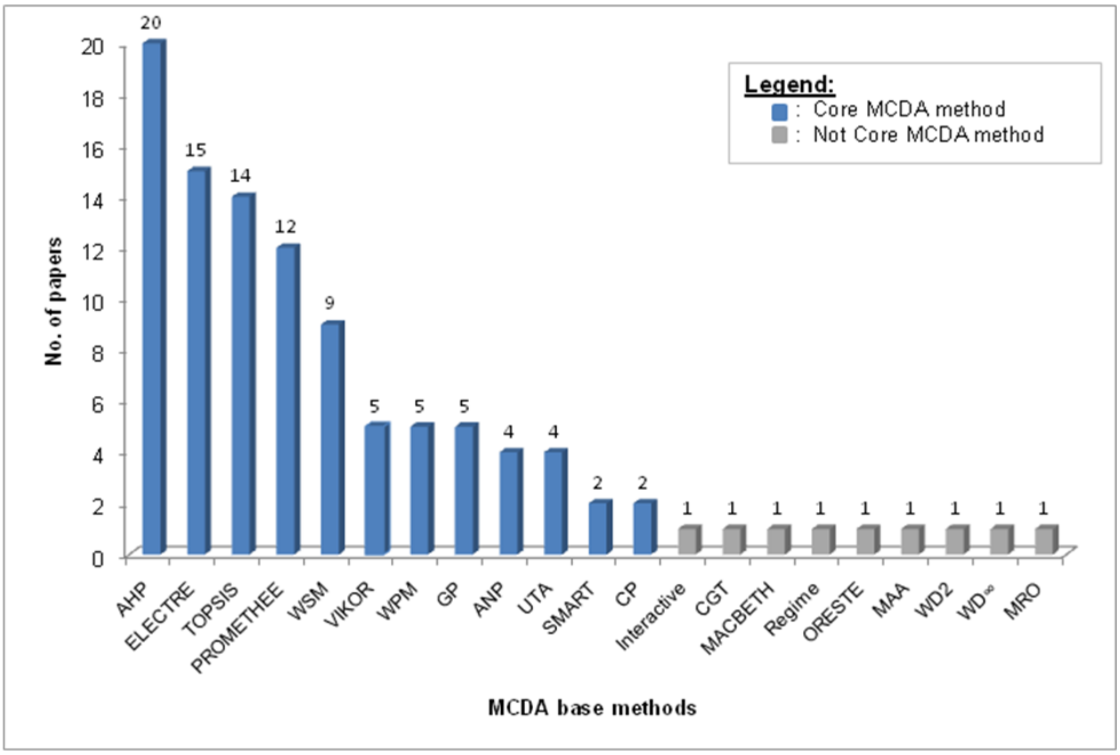

Figure 5: The MCDA base methods distribution of the publications

A total of twelve core MCDA methods are identified from this review: Analytic Hierarchical Process (AHP), Elimination and Choice Expressing the Reality (ELECTRE), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE), Weighted Sum Method (WSM), VIKOR, Weighted Product Method (WPM), Analytic Network Process (ANP), Goal Programming (GP), Utility Theory Additive (UTA), Simple Multiple Attribute Rating Technique (SMART) and Compromise Programming (CP) due to their occurrences overweigh the minimum threshold values. Table 20 summarize the basic concept of each core MCDA method which will be further analyzed in this thesis.

Table 20: Brief description of core MCDA method [14, 45]

Method	**Description**
AHP	AHP assists decision makers in deriving the family of criteria through decomposition manner. It enforces the comparative judgment on a pair of elements and transformation subjective judgment of relative importance into a set of weights through pair-wised comparison which are recombined into an overall rating of alternatives (i.e. aggregating the performance). Papers that assess AHP method are [24], [27], [43], [50], [53 - 68].
ELECTRE	ELECTRE family consists of six variations as follow: ▪ *ELECTRE I*: It applies outranking concept to aggregate the preference instead of performance. It aims to diminish the size of non-dominated set of alternatives where an alternative will be eliminated when it is dominated by another alternative to certain extent. ▪ *ELECTRE IS*: It applies exactly similar concept as ELECTRE I and yet it introduces indifference threshold into evaluation. ▪ *ELECTRE II*: It applies same concept as ELECTRE I but it introduce the usage of two outranking relations namely strong preference and weak preference. ▪ *ELECTRE III*: It's the extension of ELECTRE I to express outranking relation by using credibility index. ▪ *ELECTRE IV*: It's the extension of ELECTRE III without using any weight. ▪ *ELECTRE TRI*: It's extension of ELECTRE III to use conjunctive and disjunctive technique in order to support the ranking of alternatives. Papers that assess ELECTRE family methods are [27], [56], [57], [60], [61], [66], [67], and [69 - 76].
TOPSIS	TOPSIS evaluate the alternatives based on their distance toward the ideal solution and negative-ideal solution where the selected alternative should fall at the nearest to ideal solution but the furthest from negative-ideal solution. Papers that assess TOPSIS method are [24], [54], [56], [59-61], [66], [70-72] and [76-79].
PROMETHEE	Two variances found in this PROMETHEE family: ▪ *PROMETHEE I*: It applies the same concept as ELECTRE family but it introduces 6 functions to express preference of decision maker along each criterion. However, it provides only a partial order of the alternatives using entering and leaving flows. ▪ *PROMETHEE II*: It applies same as PROMETHEE I but it uses an aggregation of entering and leaving flows to provide total ordered of the alternatives. Papers that assess PROMETHEE family methods are [24], [27], [56], [57], [60], [61], [66], [69-71], [74] and [80].

WSM	WSM is an elementary MCDA method where the ultimate performance of alternative is assessed as the weight sum of its evaluation with respect to each criterion. Papers that assess WSM method are [27], [50], [56], [60], [66], [70], and [76-78]
VIKOR	VIKOR applies the similar concept as TOPSIS but it includes the relative importance of criteria to compute the distance of alternative from the ideal-solution and negative-ideal solution. Papers that assess VIKOR method are [70], [71], [75], [78] and [79]
WPM	WPM is an elementary MCDA method where the ultimate performance of alternative is assessed as the weight product of its evaluation with respect to each criterion. Papers that assess WPM method are [56], [66], [70], [76], and [77].
GP	GP is an approach aims to formulate a function that can optimize the objectives given by the users. It is different from the other MCDA approach as it focuses on the objectives themselves instead of the decision criteria. Papers that assess GP method are [24], [43], [61], [62], and [64].
ANP	ANP applies the same concept as AHP but it decomposes the family of criteria in the network structure where the dependencies between criteria are no longer only one direction (i.e. from upper criteria to lower criteria) but both. Papers that assess ANP method are [54], [58], [81], and [68].
UTA	UTA applies ordinal regression to estimate the value function for all criteria and acquires the ultimate value function through an additive manner. Papers that assess UTA method are [61], [63], [66], and [73].
SMART	SMART method applies the concept of multi-attribute utility theory (MAUT) that use weighted linear averages as the utility function to aggregate the performance of all criteria for each alternative. Papers that assess SMART method are [27], [66].
CP	CP evaluates the alternatives based on its distance from the predefined ideal point where the solution that has closest distance to the ideal point is interested. Papers that exemplify CP method are [56] and [65].

Discussion:

It is quite obvious from the abovementioned results that AHP method is the most common assessed method where it has been discussed or evaluated in more than 50% of the selected publications from the first review. This is may be due to its representation of a complicated decision problem into a simple hierarchical structure and its simple elicitation of the criteria weights based on the pair-wised comparison between two decision criteria on a ratio scale [61]. However its problem to have rank reversal is rarely discussed which violates the independence of alternatives of decision theory [14]. Rank reversal is referred to the case where the current ranking for any two alternatives can be reversed when a new alternative is introduced into the existing assessment [14, 61]. Though AHP method has many other issues apart from the rank reversal problem, it is still the most popular MCDA methods to support decision making in general domains. It can probably because the aforementioned advantages have compensated its disadvantages.

5.1.2 RQ 1.2: What are basic elements common for all MCDA methods?

Result & Analysis:

From the first review, the features of MCDA methods that are measured and used to compare their applicability in supporting decision making are extracted. These measurable features are regarded as the basic elements of MCDA methods. A total of forty-nine basic elements of MCDA methods are extracted from this review (refer to Table 22, Table 23, Table 24 and Table 25 for the description of each basic elements).

In order to acquire an overview of what is actually evaluated; these forty-nine basic elements are categorized according to the basic concept of MCDA [26], namely the criteria, preference evaluation and the alternative. During the course of data extraction, it is notable that the practical usage of the methods is needed to characterize the feature of MCDA methods. Therefore, a total of four concepts are used to characterize the feature of MCDA methods, namely the alternative, the preference evaluation, the criteria and the usage. These four concepts are designated as the facets of MCDA. The detail meanings of each facet are briefly illustrated in Table 22. Though the categorization scheme for these four facets is defined a-priori, it is evolved during the course of data extraction for a better characterization of the method features.

Table 21: Description of different facets for MCDA base method evaluation [14, 40]

Facets	Description	Related elements
Usage	Includes all practicality-related elements that elaborate the acceptable characteristics of the methods to justify its applicability in decision maker context.	Refer to Table 22
Criteria	Includes all technical elements that elaborate the acceptable characteristics, prerequisite or limitation of the decision criteria in order to apply the method. However, the way the preference of criteria is evaluated or aggregated are not included in this facet as this type of elements are found more related with preference handling facet.	Refer to Table 23
Alternative	Includes all technical elements that elaborate the acceptable characteristics, prerequisite or limitation of the possible action or alternatives in order to apply the method. However, this facet doesn't include the elements that relate with the way the preference of alternative is constituted as these elements are more related with preference handling facet.	Refer to Table 24
Evaluation	Includes all technical elements that elaborate the acceptable/supportable characteristics, prerequisite or limitation of the preference aggregation, preference modelling, and the presentation of the evaluation result (i.e. problematic).	Refer to Table 25

To categorize the 49 basic elements found in this review, they are grouped based on the similarity in term of their feature and the impact they can cause.

In the usage facet, eighteen basic elements are identified from the selected publications. The definition of each basic element that assesses the usage facet is summarized in Table 22. Besides, the relationship of each basic element with other basic elements are analyzed and listed in Table 22. Five of them are identified as independent element as they have no other related element. They are element *U3: Organization*, element *U5: Effort spends in decision analysis*, element *U9: Dimension of decision problem*, *U16: Implementation cost*, and element *U17: Preference of decision maker*.

Table 22: All MCDA elements that relate with method usage found in the literatures

ID	Basic elements of MCDA in usage category	Related element
U1	**Ease of use** Ease of use is understood as the degree of usability for the method from the viewpoint of users. From the review, it was found that many publications evaluate the usability of the method from three perspectives namely 1. The simplicity of the computation 2. The ease in preferential elicitation (i.e. element C5) 3. Effort spends on the evaluation (i.e. element U5).	U7, C5, U5
U2	**Uncertainty handling** Uncertainty occurs in the area of decision making can be referred as many area in which can be resulted from the imprecise measurement or quantification of criteria, trade-offs or preferences (i.e. element P). Besides, uncertainty can also cause by lacking some information needed at the time of decision is made. Therefore, the handling of uncertainty is referred to capability of MCDA method to support both imprecise measurement and evaluation under incomplete information (i.e. element U20).	U20
U3	**Organization** Organization is known as the way to structure criteria involved in order to understand the ordering, comparison or grouping within the criteria. For example, the criteria can be constructed in a tree structure to designate how they are interconnected to each other.	-
U4	**Group Decision Making** The method should provide a mechanism to elicit and aggregate the preferences from a group of individuals.	U6, U8
U5	**Effort spend in decision analysis** This element includes the total effort needed in order to structure a decision problem, elicit criteria weight, and setting the method parameter(s).	-
U6	**Judgment conflict resolution** This element assesses the capability of MCDA method to handle the contradictory judgment arises within the group of individuals involve in the decision making. More precisely, the MCDA method that supports group decision making should provide a way in certain extent for the conflicting party to assess the costs and benefits of their supporting option against the options supported by another party.	U5, U8
U7	**Tool assistance**	U1

	A software tool is existed to automate the execution of a MCDA method and ease the computation process of the MCDA method which relates with element U1.	
U8	**Applicability to group decision making [43]** The extent the group decision making can be facilitated by the MCDA method in which can be accessed from the performance of MCDA method with respect to elements from U10 until U17.	U6, U10, U11, U12, U13, U14, U15, P16, P19
U9	**Dimension of decision problem** This element is known as the number of point of view of a decision problem. Generally, all MCDA methods are handling a decision problem from multiple dimensions.	-
U10	**Leadership assurance** In the environment of group decision making, it is common to have one moderator to facilitate the whole decision analysis process. Therefore, this element is needed to study the extent the MCDA method can help the moderator to accomplish and maintain their goals.	U8
U11	**Learning** The capability to provide subjective value is assumed requiring more experience in assessing than objective value. Therefore, this element assesses the capability of MCDA methods in helping the decision groups in knowing the way to express their subjective value in the decision analysis.	U8
U12	**Scope [43]** This element aims to assess the capability MCDA method to address the abstraction of decision problem	U8
U13	**Assistance of alternative development [43]** This element does not mean searching the possible alternative for resolving a decision problem. It means that the method must provide a way for the decision group to view and identify the possible alternatives from different abstraction level. For example, the software project manager may view the problem from higher abstraction level of a decision problem than a group of software developers as they have much wider choices and this can affect their set of alternatives to be different.	U8
U14	**Priority of group members inclusion** This element covers the way the MCDA method treat the importance of each individual members of the decision group. For example, some MCDA methods treat all the preference from individual members	U8

	equally important and some allow the preference from each individual member carries different priorities.	
U15	**Fairness consideration to all stakeholder [43]** To allow fairness for all stakeholders in decision making, this element includes the MCDA method that considering the individual preferences from the stakeholders together with their priorities (i.e. element U14).	U8, U14
U16	**Implementation cost** The cost covers the expenses in implementing the MCDA method or purchasing its tool available and the expenses needed to train the decision maker to understand the MCDA method and/or using the tool. However, it is not the scope of this thesis to find out the exact cost needed but the information is relied on our subjective judgment based on the related opinions collected from the review.	-
U17	**Preference of decision maker** This element views the familiarity of the MCDA method from the view point of decision maker. It is assumed the more popular a MCDA method indicate higher preference for decision maker.	-
U18	**Incomplete input information** When the decision is made, it is common that some information needed for decision analysis may not be available. Therefore, some studies have tried to assess the MCDA method to ensure its ability to provide the evaluation result even the information needed is not complete.	U2

In the criteria facet of MCDA method, ten basic elements are found and summarized in Table 23. The interdependencies between the basic elements of criteria facet are studied and their related element(s) is identified and listed in Table 22. Two basic elements are found as independent element as there is no other element relates with them. They are element *C1: Nature of criteria* and element *C8: Dimension of criteria*. The rest of the elements are found related with each other within the criteria facet except:

- both element *C2: Optimum size of criteria* and element *C5: Interdependence between criteria* are related with element(s) from usage facet
- the element C10: Effort in criteria change is related with the element from preference facet.

Table 23: All MCDA elements that relate with method criteria found in the literatures

ID	Basic elements of MCDA in criteria category	Related Element
C1	**Nature of criteria** This element describes the data types of criteria the MCDA method supports namely qualitative data or quantitative data. Most of the methods support only one type and necessitate the quantification for the qualitative criteria data. However, a few of them are capable to support both types.	-
C2	**Optimum size of criteria** Some MCDA are not adapted to a large number of criteria as they require more effort when the number of criteria increases. Therefore, this type of MCDA methods should be accompanied by a software tool in order to reduce the considerable amount of manual work for handling larger number of criteria.	U7
C3	**Supporting criteria weight** The element shows the capability of MCDA method to include criteria weight into evaluation.	C5
C4	**Non-commensurable criteria handling** As a result of allowing multiple types of measure scale, there is a possibility to encounter incommensurable or incomparable criteria in the evaluation. Therefore, this element aims to study the treatment the method provide to overcome this issue and to which extent.	C8
C5	**Interdependence between criteria** This element discusses the capability of MCDA method in handling the interdependence relationship among criteria, such as conflicting, cooperative or independent.	U3
C6	**Nature of criteria weight** This element illustrates the data type allowed in MCDA method to express the weight or relative importance of criteria, namely qualitative or quantitative value.	C3, C5
C7	**Measure scale supported** To measure the criteria, a measure scale is needed and the type of measure scale supported depends on the nature of criteria allowed for the MCDA method. The most frequently used scale in MCDA methods are nominal and ordinal scale [32]. Some MCDA methods take criteria into consideration with multiple measure scales but others support only one type.	C1

C8	**Dimension of criteria** In practice, some criteria can provide the negative impact to the decision (or generally known as cost criteria) and another type of criteria provide positive impact to the decision (or generally known as beneficial criteria). However, most of the weighting MCDA methods treat all criteria are singleton. However, some other MCDA methods are capable to provide a way to differentiate and aggregate these two dimensions of the criteria (i.e. beneficial criteria or cost criteria) in a compromising manner.	-
C9	**Ease of criteria weight elicitation** This element assesses the simplicity of the weight elicitation methods the MCDA method supports which can be viewed from the simplicity of the computation, and the transparency of the reasoning the weight elicitation to the users.	C5
C10	**Effort in criteria change** Practically, the set of criteria to evaluate a decision problem can change from time to time. Therefore, this element intends to assess the additional effort required to cope the existing completed evaluation with the changes of criteria (i.e. the addition or deletion of criteria).	P5

The alternative facet has only four basic elements which are summarize in Table 24. Two of them are found independent, namely element *A2: Nature of alternative* and element *A3: Interdependences between alternatives*. The remaining elements are found related with other basic elements.

Table 24: All MCDA elements that relate with method alternative found in the literatures

ID	Basic elements of MCDA in alternative category	Related element
A1	**Optimum size of alternative** Some MCDA are not tailored to a huge number of alternatives and it indicates that the optimum size of alternatives for the MCDA method can be small. This is because the extra effort in handling large number of alternative can be unreasonable for those MCDA methods with small optimum size of alternatives. Consequently, a software tool is needed for them to ease the considerable manual work expected in handling large number of alternatives.	U7
A2	**Nature of alternative** This element describes the data types of alternative the MCDA method supports namely qualitative data and/or quantitative data.	-
A3	**Interdependence between alternatives**	-

	This element describes the assumption the MCDA method possess about the relationship between alternatives. Some MCDA methods assume all alternative are independent and others assume alternatives can be interrelated.	
A4	**Alternative change** In practice, the number of alternatives for a decision problem can be changed from time to time especially when the decision is amended. This can cause the introduction of new alternative(s) or elimination of the existing alternative. Consequently, this element aims to assess the additional effort needed for the MCDA method in order to incorporate the changes of alternative into the existing evaluation result.	P1

From the studies, seventeen basic elements are identified that related with the preference evaluation facet of the MCDA method and their details are summarized in Table 25. This facet has three independent elements namely element *P1: Aggregation*, *P2: Type of problematic*s and element *P13: Moment of elucidating preference.* The rest of the elements are found related with other elements from the same facet except the element P8: Rank reversal, element P11: Support relation between decision level, element *P14: Validity of evaluation*, and element *P17: "Psychophysical applicability"* [43] have the relationship with the elements from other facets.

Table 25: All MCDA elements that relate with method evaluation found in the literatures

ID	Basic elements of MCDA in preference evaluation category	Related element
P1	**Aggregation** This element illustrates the way MCDA method implicitly or explicitly combines the performance of each alternative across all criteria to compute the overall assessment of the alternative.	-
P2	**Type of problematics** This element discussed the intended way to present the evaluation result.	-
P3	**Preference model** This element discusses the preference structure of the MCDA method that designates the preference relation it supports to express preference from decision makers.	P1
P4	**Alternative evaluation** This element discusses the way the MCDA deriving the preference.	P3
P5	**Completeness** This element illustrates whether the method can provide complete	P1

	evaluation result.	
P6	**Required model parameter setting** The evaluation model of MCDA method may require additional parameter(s) (i.e. threshold) to be configured a priori.	P1
P7	**Compensation [40]** This element illustrates whether the evaluation of MCDA method allows the bad performance of a criterion to be compensated by the good performance of another criterion. However, there may be situation where some criteria are not comparable or compensable and therefore this element also indicates to which extent (i.e. total compensatory or partial or non-compensatory) the compensation is allowed in the evaluation.	P1
P8	**Rank reversal** This element assesses the robustness of the evaluation result with respect to the changes of alternative by checking whether the existing ranking or evaluation result can be reversed or changed.	A4
P9	**Constraint inclusion** This element shows whether the evaluation of alternatives can be done under a set of constraints.	P1
P10	**Risk evaluation** This element shows whether the evaluation of MCDA method take the potential risk the individual alternative possess into consideration.	P1
P11	**Support relation between decision level** This element relates with the organization of criteria in hierarchical manner and illustrates the direction of different hierarchical level (unidirectional or bidirectional) are taken into consideration for evaluation within MCDA method.	U3
P12	**Aggregation condition** This element studies whether the evaluation of MCDA method can aggregate criteria with different measure scale or nature.	P1
P13	**Moment of elucidating preference** This element indicates the time to elucidate the preference whether before or during the evaluation.	-
P14	**Validity of evaluation** This element includes the mathematical validity and generality in which formal mathematical representation of the logic and reasoning together with some assumptions are required. Besides, the higher	P16, U8

	consistency of weight assignment (i.e. P20) can help to improve the validity of the evaluation.	
P15	**Ambiguity of evaluation result** When the evaluation result from the MCDA method is partial, the decision maker can't differentiate and identify the best alternative from the evaluation result which indicates the occurrence of ambiguity.	P7
P16	**Transitivity** Transitivity relates to the preference structure the MCDA method supports. This element indicates the situation where the decision maker prefers alternative A against B and prefers alternative B against C, then it is assumed that the decision maker prefer alternative A against C.	P3
P17	"**Psychophysical applicability" [43]** This element assesses the preference model of the MCDA method whether it can support the measurement of relationship between physical factors of stimuli and the resulting sensations reflecting the decreasing response to rising stimulus which comply the Weber-Fechner law.	U8, P3

Discussion:

Most of the MCDA base methods found in this study may not be empirically reviewed or evaluated under the homogeneous environment or domain. However, we believe that this doesn't give any significant impact to group the element or evaluation aspects collected from the studies of different domain or environment. Besides, the elements found from the literature always evaluate at least two or more MCDA base methods and therefore all basic elements found from this review are the basic elements that can be used to evaluate all MCDA method, (i.e., elements that are common to all MCDA methods).

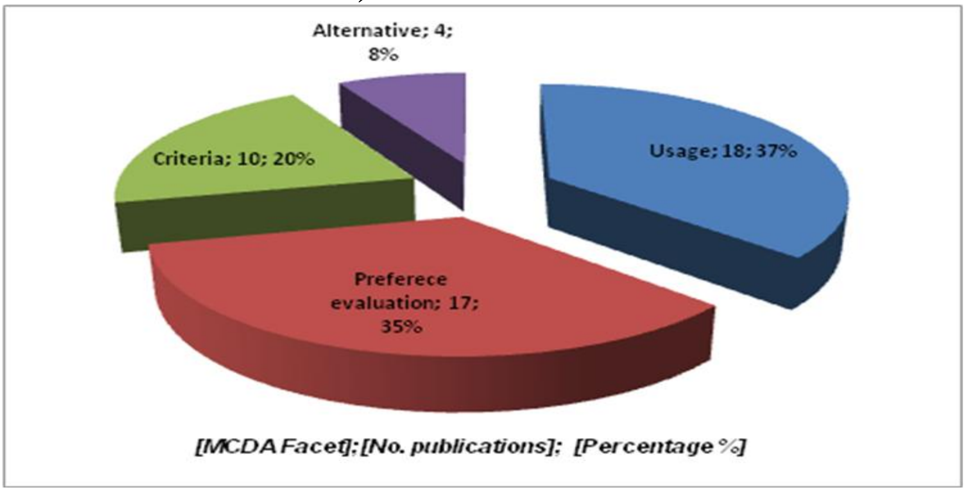

Figure 6: The distribution of basic elements to each MCDA facet.

A considerable percentage (i.e. 72%) of the total basic elements found in this study concerns both usage (36.7%) and preference (i.e. 34.7%) facets. This indicates that the MCDA facets to assess the method applicability are strongly biased towards the usage and preference facets. The dominance of these two facets and very low intensity of alternative facet indicate both usage and preference facets have higher importance as compared with the alternative facet.

5.1.3 RQ 2.1: What are the most relevant characteristics of the MCDA methods to be considered when selecting MCDA method for general decision making problems?

Result & Analysis:

From the forty-nine basic elements collected from research question RQ 1.2, the measure of each basic element is identified and then an investigation about how different extent of these basic elements can be assessed is conducted. The relevant characteristics for each basic element in this research questions is regarded as the possible value of each basic element.

Table 26 summarize the possible range of values for the basic elements collected from the publications and the respective assumption involve. Besides, Table 26 also provides the rational about the way to determine the value for each basic element. It is notable that twenty out of forty-nine basic elements carries the binary values (i.e. yes or no) which are: element *U2: Uncertainty handling*, element *U4: Group decision making*, element *U6: Judgement conflict resolution*, element *U7: Tool assistance*, element *U9: Dimension of decision problem*, element *U18: Incomplete input information*, element *C3: Support criteria weight*, element *C5: Interdependence between criteria*, element *A2: Nature of alternative*, element *A3: Interdependence between alternatives*, element *P5: Completeness*, element *P6: Required model parameter setting*, element *P8: Rank reversal*, element *P9: Constraint inclusion*, element *P10: Risk evaluation*, element P11: Support relation between decision level, element *P12: Aggregation condition*, element *P15: Ambiguity of evaluation result*, element *P16: Transitivity,* and element *P13: Moment to include preference*. The rest of the elements carry the three or more subjective values that are assigned based on the extent the MCDA method can satisfy the element.

Based on the measure of basic element, the MCDA method will be assigned a single possible characteristic by referring to the performance of MCDA method with respect to the elements. However, this is not applicable to three basic elements C7: Supported measurement scale, P3: Preference model and P2: Type of problematics where multiple value or characteristics can be assigned to a MCDA method.

As an example, the element *U1: Ease of use* is measured based on three aspects: 1. the simplicity of the computation; 2. the ease in preferential elicitation and; 3. the total estimated effort needed. It is assumed that each aspect is equally important and therefore the element U1 can be assigned one of the four possible different subjective ratings base on the number of aspects the MCDA method can satisfy. If the MCDA method is perceived to be easy to compute and required less effort in performing the evaluation starting from input preparation until the formulation of a decision, then this MCDA method will be given as medium (M) rating with respect to the element *U1: Ease of use*.

Table 26: Description about the possible characteristics of each basic element

ID	Characterization of basic elements
U1	**Ease of use** Based on these 3 perspectives and the assumption that they are equally important, we characterized this element into four types: ▪ *Low (L)* : The method fulfils none of these 3 perspectives. ▪ *Medium (M)* : The method fulfils one of these 3 perspectives. ▪ *High (H)* : The method fulfils two of these 3 perspectives. ▪ *Very High (VH)*: The method fulfils all of these 3 perspectives.
U2	**Uncertainty handling** ▪ *Yes(Y)*: The method about support the evaluation under certainty (i.e. deterministic situation) and/or the uncertainty situation which occurs as the result of limited information available or the imprecise interpretation of the information. ▪ *No (N)*: The method supports the evaluation under the deterministic situation.
U3	**Organization** ▪ *None (NA)*: The method doesn't support hierarchical structuring the decision problem in term of criteria and respective alternatives. ▪ *Tree (TR)*: The decision problem, criteria and alternative are related in a tree structure where there is only one way of communication from the upper node to the lower node. ▪ *Network (NW)*: The decision problem, criteria and alternative are presented in a network structure that allows two ways of communications between two criteria. In this structure, it is difficult to distinguish whether the hierarchical level is lower or higher, dominant or subsidiary, direct or indirect.
U4	**Group Decision Making** ▪ *Yes (Y)*: The methods able to aggregate the preference of a group of decision maker. ▪ *No* (N): The methods only allow the evaluation based on the preference of single decision maker.
U5	**Effort spend in decision analysis** Based on three areas included for effort computation as mentioned in earlier session and the assumption that they are equally important, we characterized this element into three types: ▪ *Low*: The method has fulfilled all of the following characteristics: o the numbers of method parameters that need to be configured priori are at most two. o the decision problem is analyzed by organizing its decision criteria in a hierarchical structure. o the criteria weight is elicited through subjective rating instead of objective assessment. ▪ *Medium*: The method has fulfilled two of the aforementioned characteristics.

	▪ High: The method has fulfilled only at most one of the aforementioned characteristics.
U6	**Judgment conflict resolution** ▪ *Yes (Y)*: The method able to show the consequences of each conflicting judgment by considering their relative benefit and risk. ▪ *No (N)*: The method can't let user to compare the potential consequence of the judgment make.
U7	**Tool assistance** ▪ *Yes (Y)*: A software tool exists to automate the entire evaluation process of the method. ▪ *No (NA)*: No information about its software support is found from the review.
U8	**Applicability to group decision making [43]** This element is applicable for the method that can handle group decision making (i.e. U4). Based on the number of related sub-criteria (element U6, U10, U11, U12, U13, U14, U15, P17 and P20) the method achieved and the assumption that every sub-criteria is equally important, we characterized this element into 3 types: ▪ *Low (L)*: The method is capable to fulfil less than 4 sub-criteria. ▪ *Medium (M)*: The method is capable to fulfil more than 3 but less than 7 sub-criteria. ▪ *High (H)*: The method is capable in handling more than 6 sub-criteria.
U9	**Dimension of decision problem** ▪ *Single (SD)*: The method views the decision problem from single dimension or perspectives. ▪ *Multiple (MD):* The method views the decision problem from multiple dimensions or perspectives.
U10	**Leadership assurance [43]** ▪ *Low (L)*: The method doesn't require much interaction from the group. ▪ *Medium (M)*: The method provides a structure to facilitate the group leadership. ▪ *High (H)*: The method contains some collaborative mechanisms to assist the facilitator in leading the group to achieve the consensus.
U11	**Learning** ▪ *Low (L)*: The method involves little subjective value from the decision group. ▪ *Medium (M)*: The method improves the group understanding about the cause-effect for the decision problem but not the alternatives. ▪ *High (H)*: The method improves the group understanding about the cause-effect for the decision problem and the alternatives.
U12	**Scope [43]** ▪ *None (NA)*: The method doesn't support problem abstraction. ▪ *Low (L)*: The method support problem abstraction but does not problem analysis to improve the scope of abstraction for the decision problem. ▪ *Medium (M)*: The method allows establishing the boundaries for the decision group and supports the problem analysis to improve the abstraction level of the

	decision problem. ▪ *High (H)*: The method explicitly includes double loop learning.
U13	**Assistance of alternative development [43]** ▪ *None (NA)*: The method requires the alternatives to be specified prior the evaluation. ▪ *Low (L)*: The method allows the user to formulate alternative during the evaluation but no specific way is provided. ▪ *Medium (M)*: The methods allow the development of alternative in the incremental manner ▪ *High (H)*: The methods develop the alternatives from the interaction of the decision group without specifying any requirement the alternative should satisfy (e.g. independent). ▪ *Very High (VH)*: The method guides the user to systematically develop the alternatives and require the user to specify the additional properties that the alternative should have achieved in order to validate the evaluation result.
U14	**Priority of group members inclusion** ▪ *Not applicable (NA)*: The method supports the evaluation for single individual. ▪ *Low (L)*: The method supports the evaluation to be performed based on the interest of a group of individuals where each individual is treated equally important and their interests are expressed on ordinal scale. ▪ *Medium (M)*: The method supports the evaluation to be performed based on the interest of a group of individuals where each individual is treated equally important and their ultimate interests (i.e. preference) are represented on interval scale, ratio scale or absolute scale. ▪ *High (H)*: The method allows user to specify different relative importance for each individual but it doesn't not provide a systematic way to guide user to determine the weight of the individual as a group desire. ▪ *Very High (VH)*: The method allows user to specify different relative importance of each individual and provides a systematic way to guide user to decide weight for the individuals as the group interest.
U15	**Fairness consideration to all stakeholders [43]** ▪ *Not applicable (NA)*: The method does not involve analyzing the decision problem ▪ *Low (L)*: The method involves problem analysis and allows the fairness to other stakeholders to be addressed in an implicit manner. ▪ *Medium (M)*: The method allows the fairness to other stakeholders to be addressed in explicit and qualitative manner. ▪ *High (H)*: The method allows the fairness to other stakeholders to be addressed in explicit and quantitative manner.
U16	**Implementation cost** ▪ *Low (L)*: The method is simple and can be directly implemented into software tool. ▪ *Medium (M)*: The method can be implemented into software tool with some minor configuration. ▪ *High (H)*: The method is too complex to be implemented into the software tool.

U17	**Preference of decision maker** ▪ *Low (L)*: The computation of the method is not known by the decision maker. ▪ *Medium (M)*: The decision maker has prior knowledge about the computation but reluctant to use it based on their past experience. ▪ *High (H)*: The decision maker has prior knowledge about the computation and they have high interest or strong motivated based on their past experiences in performing similar computation.
U18	**Incomplete input information** ▪ *Yes (Y)*: The method supports the evaluation even when the information needed is not complete. ▪ *No (N)*: The methods only support the evaluation when the information needed is fully complete.
C1	**Nature of criteria** ▪ *Qualitative (QL)*: The method allows only qualitative value to be assigned to all criteria. ▪ *Quantitative (QT)*: The method allows only quantitative value to be assigned to all criteria. ▪ *Mixed (MX)*: The method allows either quantitative or qualitative value for all criteria. (e.g. Outranking method)
C2	**Optimum size of criteria** ▪ *Low (L)*: The process to execute the method will get exponentially onerous when large numbers of criteria is involved. ▪ *Medium (M)*: The process to execute the method requires reasonable effort (in linear way) when the number of criteria is getting large. ▪ *High (H):* The process to execute the method do not get onerous for the case when the number of criteria getting smaller or larger.
C3	**Supporting criteria weight** ▪ Yes (Y): The method includes the relative importance of the criteria into evaluation. ▪ No (N): The method treats all criteria equally important.
C4	**Non-commensurable criteria handling** ▪ *Low (L)*: The method can only handle commensurable criteria. ▪ *Medium (M)*: The method allows criteria to be specified in non-commensurable scale but scales transformation is required. ▪ *High (H)*: The method requires neither the single scale enforcement nor the scales transformation. The method is capable to aggregate non-commensurable criteria and develop the ultimate evaluation result.
C5	**Interdependence between criteria** ▪ *Yes (Y)*: The method considers the interdependence relationship between the criteria into the evaluation. ▪ *No (N)*: The method does not consider the interdependence relationship between the criteria into the evaluation.
C6	**Nature of criteria weight in the MCDA method** ▪ *Crisp number (CN)*: The method expresses the weight of criteria in numerical

	value. ▪ *Non-crisp number (QL)*: The method expresses the weight of criteria in non-numerical value.
C7	**Supported measurement scale** ▪ To measure the criteria value, the method can support one or many of the measure scale, namely nominal scale (NS), ordinal scale (OS), interval scale (IS), ratio scale (RS), and absolute scale (AS).
C8	**Dimension of criteria** ▪ *Single dimension (SD)*: The method treats all the criteria formulated from the same dimension. ▪ *Multiple dimensions (MD)*: The method treats all the criteria can be different dimension in which the good performance of one criterion can give negative impact to another criterion.
C9	**Ease of criteria weight elicitation** ▪ *Low (L)*: The method considers the criteria weight into evaluation but doesn't provide any explicit mechanism to elicit criteria weight and therefore its elicitation is done arbitrary. ▪ *Medium (M)*: The method provides an explicit but complex mechanism to elicit the criteria weight to be use in the evaluation. ▪ *High (H)*: The method provides and explicit and simple mechanism to elicit the criteria weight to be used in the evaluation.
C10	**Effort in criteria change** ▪ *Low (L)*: The addition or deletion of the criteria can be reflected into the existing computation without incurring additional effort. ▪ *Medium (M)*: The addition or deletion of the criteria can be reflected into the existing computation with minor adjustment. ▪ *High (H)*: The existing computation cannot be reused and the entire computation has to be re-executed to cater the addition or deletion of the criteria.
A1	**Optimum size of alternative** ▪ *Low (L)*: The process to execute the method will get exponentially onerous when large numbers of alternative need to be consider. ▪ *Medium (M)*: The process to execute the method requires reasonable extra effort (in linear manner) to handle larger set of alternatives. ▪ *High (H)*: The process to execute the method does not get onerous when large number of alternatives involve.
A2	**Nature of alternative** ▪ *Discrete (DS)*: The method handles a finite number of alternatives. ▪ *Continuous (CT)*: The method handles an infinite numbers of alternatives.
A3	**Interdependence between alternatives** ▪ *Yes (Y)*: The method agrees about the existence of the interdependence relations between the alternatives and takes this into consideration for its assessment. ▪ *No (N)*: The method treats all alternative are mutually exclusive.

A4	**Alternative change** ▪ *Low (L)*: The addition or deletion of the alternative can be reflected into the existing computation without incurring additional effort. ▪ *Medium (M)*: The addition or deletion of the alternative can be reflected into the existing computation with minor adjustment. ▪ *High (H)*: The existing computation cannot be reused and the entire computation has to be re-executed to cater the addition or deletion of the alternative.
P1	**Aggregation** ▪ *Single Synthesizing criterion (SS)*: This approach intends to formulate an aggregation function to indicate the global preference of decision makers for each alternative. ▪ *Outranking (OR)*: This approach allows non-compensatory criteria and capable in handling incomparability between alternatives by incorporating more preference structures.
P2	**Type of problematics** ▪ *Sorting (*β*)*: The evaluation results are grouped into a list of predefined clusters or classes. ▪ *Choice (*α*)*: The evaluation results are presented as a list of options for the decision maker to select. ▪ *Ranking (*γ*)*: The evaluation results are presented in a specific order.
P3	**Preference model** The preference structure of the method assumes one or more of the following properties (refer to Table 9 in Section 2.4.3 for definition): ▪ *Strict Preference (SP)* ▪ *Indifference (ID)* ▪ *Weak preference (WP)* ▪ *Incomparability(IC)*
P4	**Alternative evaluation** ▪ *Tradeoffs (TO)*: This approach requires the decision maker to specify the amount of a criterion, c_j, which can be compensated in order to obtain better increment in another set of criteria, c_i, for all criteria. (i.e. all values of i and j) ▪ *Direct Rating (DR)*: This approach is used to evaluate the criteria scoring performance for each alternative when there is no universally agreed scale of measurement for the criterion or lacking of resources to carry out the measurement. It uses subjective judgment from an expert to associate a number between the 0 and 100 based with the value of each option on a criterion. ▪ *Pair-wise comparison (PW)*: This approach evaluates the scoring performance of alternative with respect to a criterion indirectly by verbally compare a pair of criterion (i.e. pair-wise comparison) at a time. ▪ *Lotteries (LT)*: This approach requires a utility function that translating measure of accomplishment on the criterion considered into a value score.
P5	**Completeness** ▪ *Yes (Y)*: The method provides complete evaluation.

	▪ *No (N)*: The method provides partial evaluation.
P6	**Required model parameter setting** ▪ *Yes (Y)*: The evaluation model requires some model parameter need to be configured. ▪ *No (N)*: The method does not require any additional model parameter.
P7	**Compensation [40]** ▪ *High:* The total compensatory MCDA method that allows the absolute compensation between the different criteria. ▪ *Medium:* The partial-compensatory MCDA method that allows certain kind of compensation (not absolute compensation) between the different criteria. ▪ *Low:* The non-compensatory MCDA method where no compensation between different criteria is accepted.
P8	**Rank reversal** ▪ *Yes (Y)*: The addition of new alternative may change the existing ranking evaluation result (i.e. reverse the existing ranking). ▪ *No (N)*: The addition of new alternative doesn't change the order of the existing ranking evaluation.
P9	**Constraint inclusion** ▪ *Yes (Y)*: The evaluation of MCDA method is able to consider the constraint of achieving the goal or resolving the decision problem. ▪ *No (N)*: The evaluation of MCDA method is not able to include the constraint of achieving the goal or resolving the decision problem.
P10	**Risk evaluation** ▪ *Yes*: The evaluation of MCDA method includes the possible risks of implementing or selecting each alternative. ▪ *No*: The evaluation of MCDA method excludes the possible risk of implementing or selecting each alternative.
P11	**Support relation between decision level** ▪ *Yes (Y)*: When criteria are structured in hierarchy, the evaluation includes the relationship from the upper criteria (i.e. parent element) to their sub-criteria (i.e. child element) and vice versa. ▪ *No (N)*: When criteria are structured in hierarchy, the evaluation includes only the relationship from the upper criteria (i.e. parent element) to their sub-criteria (i.e. child element).
P12	**Aggregation condition** ▪ *Yes(Y)*: The method assumes all criteria can be aggregated. ▪ *No (N)*: The method does not assume all criteria can be aggregated.
P13	**Moment to include preference** ▪ *A priori (AP)*: The method elucidates the preference directly. ▪ *Posterior (PT)*: The method elucidates the preference indirectly.
P14	**Validity of evaluation** ▪ *Low (L)*: The method doesn't require any mathematics in computation.

	▪ *Medium (M)*: The method requires mathematics in computation but it is not generalized with the help of axiom. ▪ *High (H)*: The method requires computation in mathematics and axiom. ▪ *Very high (VH)*: The method's computation is axiom and generalized in the way that new assumption is no longer required.
P15	**Ambiguity of evaluation result** ▪ *Yes (Y)*: The method provides complete evaluation and the decision maker able to differentiate the variance of the alternatives *No (N)*: The method does not provide complete evaluation.
P16	**Transitivity** ▪ *Yes (Y)*: The method assumes its preference structure fulfils the transitivity property. ▪ *No (N)*: The method does not assume its preference structure fulfils the transitivity.
P17	"**Psychophysical applicability" [43]** ▪ *Low (L)*: The method doesn't support the analysis in the stimulus-response manner ▪ *Medium (M)*: The method supports decision problem analysis in stimulus-response manner but it needs a complex model (which may not be practical to develop). ▪ High (H): The method supports decision analysis with an effective and psychophysically applicable model.

To identify which are the most relevant characteristic of MCDA method to be considerred in decision making, the frequency of each basic elements being discussed in the selected publications are extracted. The higher frequency the basic element possess indicate the more relevant to have its characteristics to be considered in the decision making. Due to the considerable number of basic elements, their distribution of publications are illustrated separately in accordance to the MCDA facet where Figure 7 is illustrated for the basic elements of usage facet, Figure 8 is illustrated for the basic elements of criteria facet, Figure 9 is illustrated for the basic elements of preference evaluation facet and Figure 10 is illustrated for the basic elements of alternative facet.

Among all the select publications, nearly 50% of them have considered the element *C7: Supported measurement scale* and this make its characteristics to be the most relevant characteristics to be considered when selecting MCDA method for general decision making problems (i.e. 19 publications). It is followed by the element *P4: Alternative evaluation* (i.e. 16 publications) and both element P1: Aggregation and P2: Types of problematics shares the same occurrence (i.e. 14 publications).

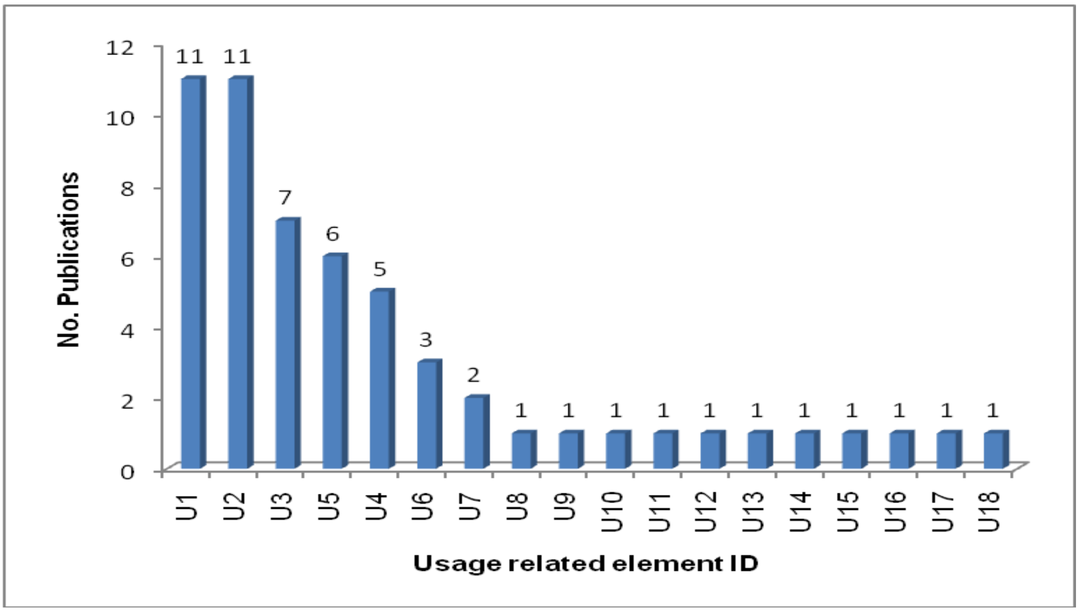

Figure 7: The distribution of publications for basic elements in usage facet

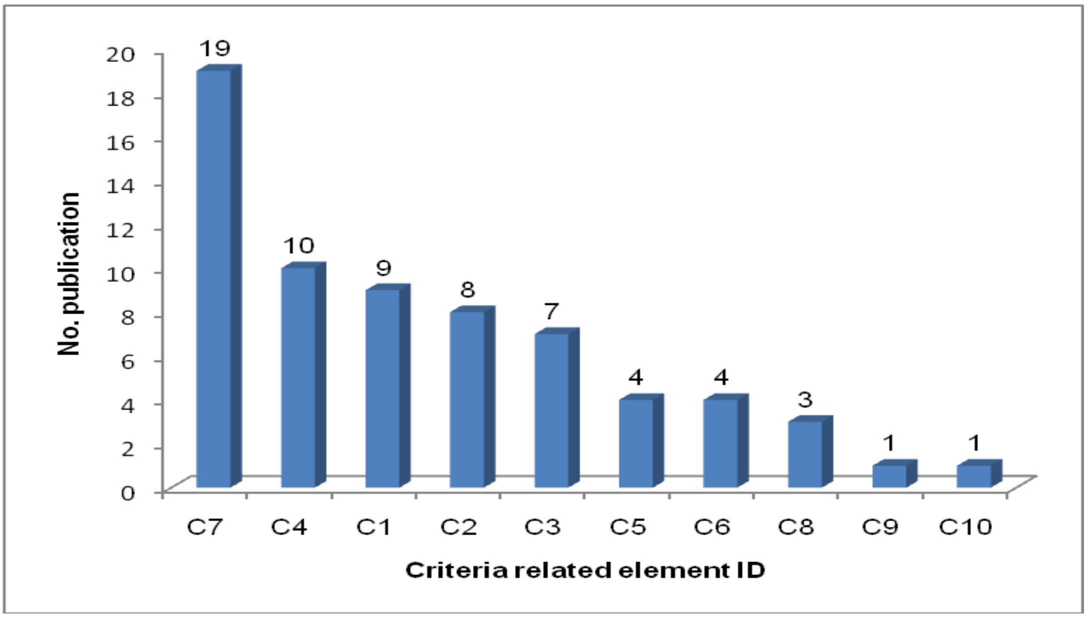

Figure 8: The distribution of publications for basic elements in criteria facet

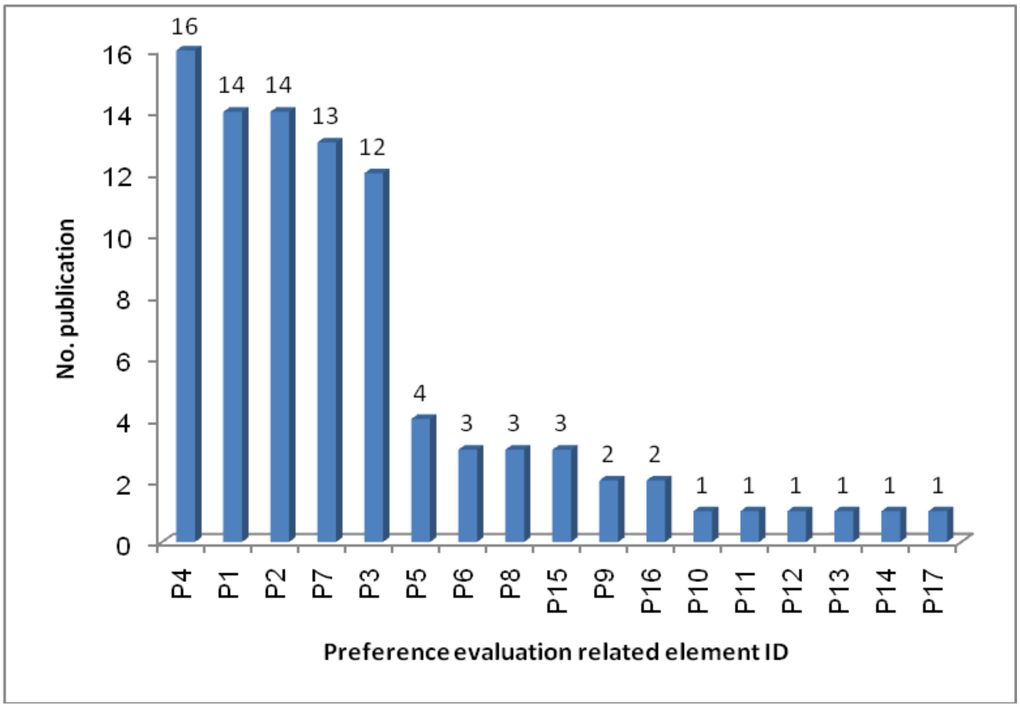

Figure 9: The distribution of publications for basic elements in preference evaluation facet

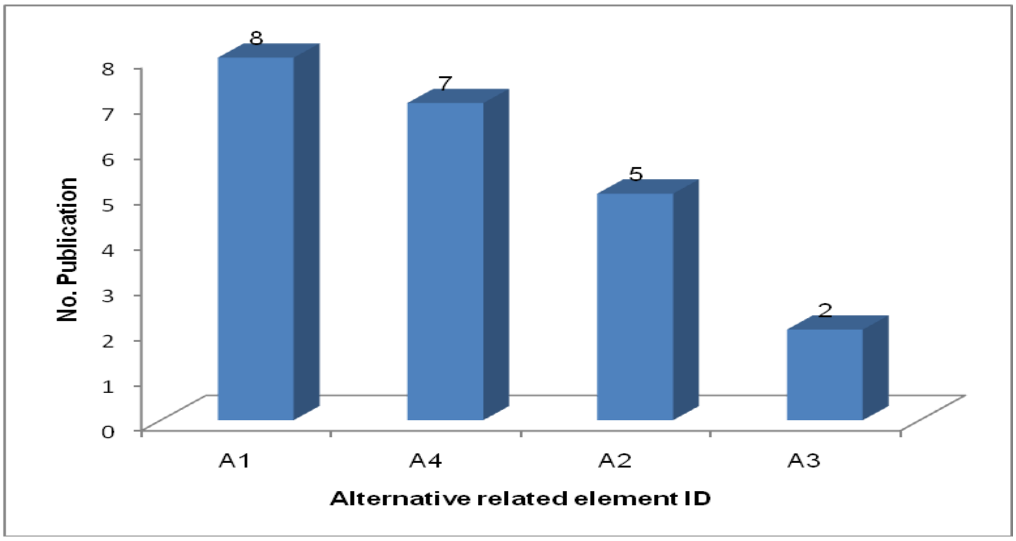

Figure 10 : The distribution of publications for elements in alternative facet

Discussion:

From the result, it is observed that the main incentive behind the assessment of an MCDA method is to help in finding a suitable MCDA method that has the maximum fitness and minimal non-fitness with respect to the need and constraints from a specific application domain. To make the choice of the suitable MCDA method, it is crucial to firstly understand and compare the capabilities of each MCDA methods from different features (i.e. the basic element).

Therefore, a comparative analysis is conducted to study the behaviours of 11 core MCDA methods collected from research question RQ 1.1. Each core MCDA method is assessed with respect to forty-nine basic elements found in this review. Based on the rational given in Table 26, the performance of the method are determined by the judgement collected from the publications and/or our subjective understanding. The overall comparative analysis results for these twelve core MCDA methods are summarized in Table 27, Table 28, and Table 29. The rows of the comparative result tables (i.e. Table 27, Table 28, and Table 29) contain twelve core MCDA methods; while their columns contain the forty-nine basic elements found in this review. The values in the cells designate the extent the MCDA method is possessed with respect to the respective basic element which is represented in the form of abbreviations. The definition of each abbreviation has been defined earlier in Table 26.

As an example from Table 27, some comparative results of AHP method is described to give a better illustration about how the comparative analysis is conducted. Firstly, AHP is rated high (abbreviation: 'H') with respect the element *U1: Ease of Use* due to:

- its computation procedure is perceived simple that is exemplified by [27], [50], [55], and [74].
- its paired-wise comparison make the preferential information easier to be elicited that is exemplified [27].

Second, AHP method does not support nine basic elements and therefore AHP method is assigned 'no' (abbreviation 'N') with respect to all these nine elements that indicate its incapability. The nine basic elements are element *U2: Uncertainty handling*, element *U18: Incomplete input information*, element *A3: Interdependence between alternatives*, element *P6: Required model parameter setting*, P8: Constraint evaluation, element *P10: Risk evaluation*, element *P11: Support relation between decision level*, element *P15: Ambiguity*, and element *P16: Transitivity*. Conversely, AHP method has satisfied nine basic elements and therefore it is assigned as 'yes' with respect to these nine elements: U4: Group decision making, element U6: Judgement conflict resolution, U7: Tool assistance, C3: Supporting criteria weight, C4: Non-commensurable criteria handling, element C5: Interdependence between criteria, element P5: Completeness, element P8: Rank reversal, and P13: Aggregation condition. However, it is noted that not all fulfilment of the basic elements give a positive impact. The fulfilment of AHP method with respect to element P8: Rank reversal makes it to be disadvantage in assessment as the reliability of AHP method can be questionable due to the possible change of ranking.

Table 27: Comparative analysis result of the core MCDA methods (usage perspective)

MCDA Method	Usage Perspective																	
	U1	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**U8**	**U9**	**U10**	**U11**	**U12**	**U13**	**U14**	**U15**	**U16**	**U17**	**U18**
	Ease of use	Uncertainty handling	Organization	Group Decision Making	Effort spend in decision analysis	Judgment conflict resolution	Tool assistance	Applicability to group decision making	Dimension of decision problem	Leadership assurance	Learning	Scope	Assistance of alternative development	Priority of group member inclusion	Fairness consideration to all stakeholder	Implementation cost	Preference of decision maker	Incomplete input information
AHP	H	N	TR	Y	M	Y	Y	H	M	H	VH	M	VH	VH	H	M	H	N
ELECTRE	M	Y	NA	N	H	N	Y	M	M	M	H	M	H	H	L	H	H	Y
TOPSIS	M	N	NA	N	M	Y	NA	L	M	M	M	M	H	NA	L	M	M	N
PROMETHEE	H	Y	NA	Y	M	N	Y	M	M	M	H	M	H	H	L	H	H	Y
WSM	VH	N	NA	N	L	N	NA	L	M	L	M	L	NA	NA	NA	L	M	N
VIKOR	M	Y	NA	N	M	Y	NA	L	M	M	H	M	H	NA	L	H	L	N
WPM	VH	N	NA	N	L	N	NA	L	M	L	M	L	NA	NA	NA	L	M	N
ANP	L	N	NW	Y	M	Y	NA	H	M	H	VH	M	VH	VH	H	M	L	N
GP	M	N	NA	N	M	N	NA	M	M	L	L	M	L	NA	L	M	M	Y
UTA	M	N	NA	Y	H	N	Y	H	M	L	M	M	H	M	M	M	L	N
SMART	H	N	NA	Y	H	Y	Y	H	M	M	H	M	H	H	M	M	M	N
CP	M	N	NA	N	H	Y	NA	L	M	M	M	M	VH	NA	L	H	L	N

**the definition of the abbreviations used is referred to Table 26.*

Table 28: Comparative analysis result of the core MCDA methods (criteria & alternative perspectives)

MCDA Method	Criteria Perspective														Alternative			
	C1	C2	C3	C4	C5	C6	C7					C8	C9	C10	A1	A2	A3	A4
	Nature of criteria	Optimum size of criteria	Supporting criteria weight	Non-commensurable criteria handling	Interdependence between criteria	Nature of criteria weight	Supported measurement scale					Dimension of criteria	Ease of criteria weight elicitation	Effort in criteria change	Optimum size of alternative	Nature of alternative	Interdependence between alternatives	Alternative change
							NS	OS	IS	RS	AS							
AHP	MX	L	Y	Y	Y	CN	N	N	Y	N	N	SD	H	H	L	DS	N	H
ELECTRE	MX	L	Y	Y	N	CN	N	Y	N	N	N	MD	L	M	H	DS	Y	L
TOPSIS	QT	H	N	N	N	CN	N	N	Y	N	N	MD	L	M	H	DS	N	M
PROMETHEE	MX	H	Y	Y	N	CN	N	N	Y	N	N	SD	L	M	H	DS	Y	L
WSM	QT	H	Y	Y	N	CN	N	N	Y	N	N	SD	L	L	VH	DS	N	M
VIKOR	MX	H	Y	N	N	CN	N	N	Y	N	N	MD	L	M	H	DS	N	M
WPM	QT	H	Y	Y	N	CN	N	N	Y	N	N	SD	L	L	H	DS	N	M
ANP	MX	L	Y	Y	Y	CN	N	N	Y	N	N	SD	H	H	L	DS	N	H
GP	QT	M	Y	N	N	CN	N	N	Y	N	N	NA	NA	L	H	CT	N	M
UTA	MX	H	N	Y	N	CN	N	N	Y	N	N	SD	L	M	M	DS	N	M
SMART	MX	L	Y	Y	N	CN	N	N	Y	N	N	SD	M	M	M	DS	N	M
CP	QT	L	Y	N	N	CN	N	N	Y	N	N	SD	L	M	H	DS	N	M

**the definition of the abbreviations used is referred to Table 26.*

Table 29: Comparative analysis result of the core MCDA methods (preference perspective)

MCDA Method	Preference Perspective																					
	P1	**P2**			**P3**				**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**
	Aggregation	Type of problematics			Preference Model				Alternative evaluation	Completeness	Required model parameter setting	Compensation	Rank reversal	Constraint evaluation	Risk evaluation	Support relation between decision level	Aggregation condition	Moment to include preference	Validity	Ambiguity	Transitivity	Psychophysical applicability
		α	β	γ	SP	WP	ID	IC														
AHP	SS	Y	N	Y	Y	N	Y	N	PW	Y	N	M	Y	N	N	N	Y	AP	H	N	N	VH
ELECTRE	OR	Y	Y	Y	Y	Y	Y	Y	PW	N	Y	M	N	Y	N	N	N	AP	M	Y	N	M
TOPSIS	SS	Y	N	Y	Y	N	Y	N	DR	Y	Y	M	Y	Y	N	N	Y	AP	M	Y	Y	M
PROMETHEE	OR	N	Y	Y	Y	N	Y	Y	PW	Y	Y	M	Y	Y	Y	N	N	AP	M	N	N	M
WSM	SS	Y	N	Y	Y	N	Y	N	DR	Y	N	H	N	N	N	N	Y	AP	H	N	Y	L
VIKOR	SS	Y	N	Y	Y	N	Y	N	DR	Y	Y	M	Y	Y	Y	N	N	AP	M	N	Y	M
WPM	SS	Y	N	Y	Y	N	Y	N	DR	Y	N	H	Y	N	N	N	Y	AP	H	N	Y	L
ANP	SS	Y	N	Y	Y	N	Y	N	PW	Y	N	M	Y	N	N	Y	Y	AP	H	N	N	VH
GP	NA	Y	Y	Y	Y	N	Y	N	TO	N	N	M	N	Y	N	N	Y	AP	M	N	N	M
UTA	SS	Y	Y	Y	Y	N	Y	N	TO	Y	N	M	N	Y	N	N	Y	AP	M	N	Y	M
SMART	SS	Y	Y	Y	Y	N	Y	N	TO	Y	N	M	Y	N	N	N	Y	AP	H	N	Y	M
CP	SS	Y	N	Y	Y	N	Y	N	DR	Y	Y	M	Y	Y	Y	N	N	AP	M	N	Y	M

**the definition of the abbreviations used is referred to Table 26.*

5.1.4 RQ 2.2: What are the most relevant requirements regarding software quality assessment?

Result & Analysis:

In the second review, the factors and issues that influence and restrict the work in software quality assessment are extracted from the selected publications and then an investigation on the affinity of each requirement is conducted. The tendency of each requirement designates as the extent of relevancy for each identified requirement in software quality assessment which are measured based on its occurrence.

In total, twenty requirements for the quality assessment method are found from the second review and the details of each requirement are described in Table 30.

To justify the relevancy of these twenty software quality assessment requirements, its numbers of publications are investigated and the details of the requirements distribution of the publications are illustrated in Figure 11. As shown in Figure 11, the requirement Q12 (i.e. quality trade-offs) has the highest relevancy with 10 publications and it is followed by requirement Q14 (i.e. uncertainty handling) with 9 publications. However, both requirement Q13 and Q20 are sharing the lowest number of publications (i.e. one publication) and therefore they have the lowest relevancy.

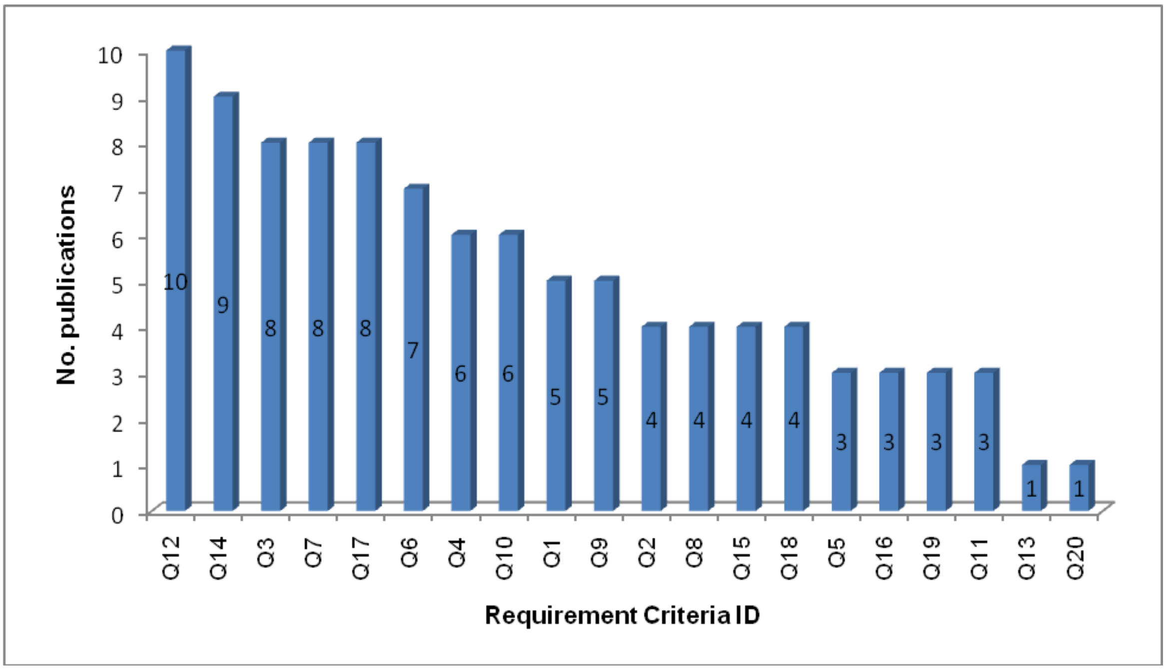

Figure 11: Distribution of quality assessment requirements found in the review

Table 30: Description of the SQA requirements

SQA requirement	Description
Q1: The method shall be comprehensable.	As suggested in [45], the method's comprehensibility relies on : ▪ the simplicity of its underlying theory ▪ the simplicity of assessing the input data ▪ the previous knowledge with the method ▪ the time require to understand the assessment Papers that exemplify this requirement are [82-86].
Q2: The method shall be able to aggregate different quality attributes.	The assessment model of the method is capable to combine all quality measures of a software product to present the overall software quality. Papers that exemplify this requirement, Q2, are [25], and [86-88].
Q3:The method shall support group decision making.	To support group decision making, the method should be capable to asseess the group interest with the quality aspects and resolve the conflicting judgement in the assessment. Papers that exemplify this requirement, Q3, are [9], [49], [51], [86], and [88-91]
Q4: The method shall be flexible in modeling the quality.	In the real-industry environment, the hierarchy of the quality aspects can be varied according to the software product and also the application context. Therefore, this requirement refer to the capability of the method to intergrate and correlate different structure of the quality models. Papers that exemplify this requirement, Q4, are [51], [83], and [92-95].
Q5: The method shall include the consideration about the values and consequences of applying the software product.	Once a software product is selected, its actual usage can deliver certain business value and carry some impacts or consequences to the application area. Therefore, the assessment theory model of the method should take both of these two criteria into consideration. Papers that exemplify this requirement, Q5, are [90], [92], and [96]
Q6: The method shall be able to perform evaluation even the information required is not complete.	It is a common problem in the software industry where the required input data for assessment is not available or incomplete at the time of actual assessment. Therefore, the quality assessment method should be able to provide reliable evaluation result even under the case where the

	required input data is incomplete or limited. Papers that exemplify this requirement, Q6, are [94], and [96-101]
Q7: The method shall be sufficiently robust to the size of software product.	Nowadays, both of the size and complexity of the software product is getting larger. It is assumed that the more complex or larger the software product, the more quality aspects have to be considered. Therefore, the quality assessment method should be robust towards the number of quality attributes in certain extent where its aplication effort required shall not exponentially increase for larger and complex software product. Papers that exemplify this requirement, Q7, are [48], [49], [83], [88], [93], [96], [102], and [103].
Q8: The method shall provide an unambigious computation procedure which all decision makers should understand it in the same way.	The computation procedure should be explicit so that the stakeholders can understand in the same way and consistently provide their preferences based on the performance of each quality attribute. Papers that exemplify this requirement, Q8, are [49], [87], [103], and [104].
Q9: The method shall have repository to store evaluation scores for future reuse purpose.	By storing the assessment results into the repository, the stakeholders can easily retrieve and access them for reference wihout any constraint. This also allow the evaluation result to be reused for the similar software product in the future and therefore it may reduce the effort in assessing the similar software product with minor adjustment as compare with performing the complete assessment from scratch. Papers that exemplify this requirement, Q9, are [85], [94], and [103-105],
Q10: The method shall require less assessment effort.	The assessment effort is the total efforts expected spend in assessing the input data, configuring the model parameters and modeling the quality. Papers that exemplify this requirement, Q10, are [82], [86], [93], [102], [105], and [106].
Q11: The method shall provide a common benchmark system.	Different software products can be assessed in different measurements and this makes the comparison between their assessed quality difficult. Therefore, the assessment method should provide a common benchmarking system that allowing the software products to be compared via the common measures. Papers that exemplify this requirement, Q11, are [106],

	[107], and [108]
Q12: The method shall facilitate quality trade-offs.	The required quality aspects for the software products can be contradictory to each other in which the better performance on a quality aspect can give a negative/positive impact to another quality aspect. This requirement is evaluated as the degree of trade-off between the quality factors allowed in the underlying assessment theory of the method. Papers that exemplify this requirement, Q12, are [49], [86], [88], [89], [91], [98], [101], and [106-108]
Q13: The method shall be repeatable.	This requirement refers to the repeatability of the method, in particular the additional efforts in handling the changes are expected to be low. Paper that exemplifies this requirement, Q13, is [82].
Q14: The method shall be able to handle uncertain data.	In the real world, the assessment criteria are not always deterministic and it is possible to have uncertain criteria. This requirement refer to the capability of the method to express the input data with non-deterministic value. Papers that exemplify this requirement, Q14, are [25], [49], [51], [88], [91], [100-102], and [109]
Q15: The method shall be able to handle interdependencies between the quality attributes.	This requirement refer to the inclusion the relationship between the quality criteria to the method's underlying assessment model. Papers that exemplify this requirement, Q15, are [25], [35], [48], and [51].
Q16: The method shall have been implemented in a software tool to assist or automate its manual computation.	To access software quality, the process to obtain the relationship among dimensions and attributes may require exhaustive time and effort. This requirement refers to the automation of the method from: ▪ the help of an existing software tool that has implemented the assessment theory of the method ▪ the new software implementation for the method with reasonable implementation cost. Papers that exemplify this requirement, Q16, are [35], [103], and [110].
Q17: The method shall express the reliability of the stakeholders' opinions in quantifiable and explicit manner.	In practise, every stakeholders have different responsibility and this make the impact of their opinion in the evaluation to be varied. Therefore, this requirement refers to the way the method

	express the priority of stakeholders and to which extent. Papers that exemplify this requirement, Q17, are [25], [83], [85], [91], [92], [93], [103], and [108].
Q18: The method shall express the variance of the judgements from the stakeholders in explicit and quantifiable manner.	The assessment of software quality leads to a decision to the choice of a software product. The selected software product can affect to the stakeholders differently and therefore their interpretation of the same measure result of the quality aspects or quality attributes can be varied. Therefore, this requirement elaborates the capability of the method to include these variances and to which extent. Papers that exemplify this requirement, Q18, are [84], [90], [92], and [101].
Q19: The method should be applicable for the standard quality model.	To be applicable to the quality model, the method should organize the decision criteria in a hierarachical structure in particular the tree-like structure as it is common required in the standard quality models. For example ISO 9126. Papers that exemplify this requirement, Q19, are [86], [87], and [103].
Q20: The method shall provide reliable assessment result.	This requirement refers to the validity and consistency of the method's assessment result. Paper that exemplifies this requirement, Q20, is [82].

The requirements *Q7: The method shall be sufficiently robust to the size of software product* is found not so strongly relevant to the SQA method. This is because the requirement Q7 is more related in providing a base for the measure of the quality attributes. The requirement Q7 shall have no impact on the software quality assessment method. Therefore, it is removed from the list and make the number of requirements from this review to be nineteen.

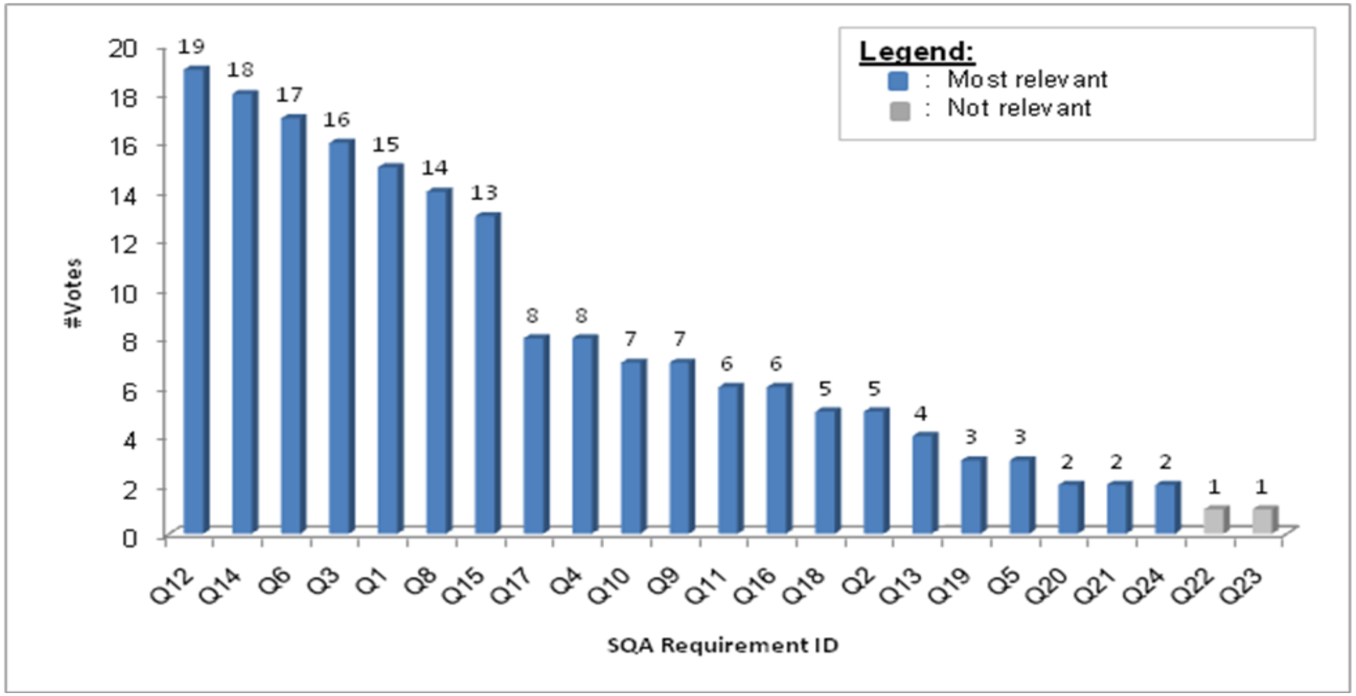

Figure 12: Distribution of Quality Requirements after consolidated with the industrial survey

Discussion:

To strengthen the result of this review, nineteen relevant requirements gathered from this review are consolidated with the results from an industrial survey [47]. This industrial survey [47] is a questionnaire results about the subjective rating of a given set of requirements regarding software quality assessment methods which are related to the work of this second review.

Nineteen requirements collected from the second review are mapped to the requirements mentioned in the industrial survey by comparing their definition and the area of impact. Through the course of the requirement mapping, four new requirements are introduced as they are not found from this review but have been mentioned in the industrial survey [47]:

- Q21. The method shall support software improvement.
- Q22. The method shall able to handle incorrect input.
- Q23. The method shall support the threshold setting.
- Q24. The method shall facilitate the learning mechanism

This makes the number of SQA requirements increase to twenty-three. The ranking of each SQA requirement are evaluated based on the total votes collected from the second review and the survey [47]. The vote here is regarded as the number of publications or the number of companies selecting the requirement. It is our assumption that the vote from the publication and the industrial survey are equally important. Based on this assumption, the rankings of the SQA requirement are formulated in accordance to the sum of the votes and the details are illustrated in Figure 12.

By comparing the results shown in Figure 11 and Figure 12, the top two requirements remain the same as the review, namely the quality attributes trade-off (i.e. requirement Q12) and the handling of data uncertainty (i.e. requirement Q14). Among the requirement with the ranking change, the requirement Q1 has the highest changes where it is initially ranked as 9th from the second review and then changed to 5th position after consolidating the result from the industrial survey.

To determine whether the SQA requirements are the most relevant, we evaluate based on its high frequency from the review and the survey. As mentioned earlier, twenty-three SQA requirements are found and they are assumed equally important. Therefore, the threshold value for high frequency is defined as at least equal to the average of percentage shared among them, i.e. approximately 4.34% or 2 votes. Consequently, the requirement Q22 and Q23 are classified as non-relevant because their single vote is less than the threshold value (i.e. 2 votes).

In short, twenty-one SQA requirements are found as the most relevant regarding software quality assessment and summarized in Table 51 (refer to Section 0). The distributions of these twenty-one requirements are illustrated in Figure 12. It is notable that this figure designates the studies where a common SQA requirement was mentioned as standalone and also in conjunction with other SQA requirements. A substantial percentage of the number of votes concerns the requirements to support the quality trade-off (i.e. requirement Q12, 42%) and input data which carries non-deterministic value (i.e. requirement Q14, 40%).

6 PROPOSED SELECTION FRAMEWORK

The aim of this chapter is to illustrate a selection framework that we propose and its concept that provide the base to analyze and select suitable MCDA methods for software quality assessment (SQA). The result of systematic review (i.e. Chapter 6) covers a broad field of MCDA methods used in general decision making and the user expectations towards the software quality assessment methods. The main finding from the reviews reveals the lack of a systematic approach to evaluate the suitability of MCDA methods in the context of software quality assessment.

By analyzing and interpreting the information collected from both reviews, the building blocks of the framework are conceptualized in order to propose a general approach to method suitability evaluation in software quality assessment. To support the proposed framework, two intermediate results collected from the systematic reviews are directly reused:

- the basic elements of MCDA method are served as the base of the MCDA method suitability evaluation
- the collected SQA requirements (refer to Section 5.1.4) is served as the reference for the practitioners to formulate their requirement about software quality assessment method

The focus of the proposed framework is to assist the software practitioners to define their expectation towards software quality assessment, to select suitable MCDA methods with respect to the user expectations and to seek the adjustment needed to apply the selected approach. Figure 13 illustrates the overview of the MCDA-SQA selection framework and the respective input and outputs. The MCDA-SQA selection framework consists of six main steps and each step is described in the following sections:

1. Characterization of the application domain (Section 6.1)
2. Methods elimination (Section 6.2)
3. Generalization to decision situation (Section 6.3)
4. Comparative analysis (Section 6.4)
5. Assessment (Section 6.5)
6. Refinement (Section 6.6)

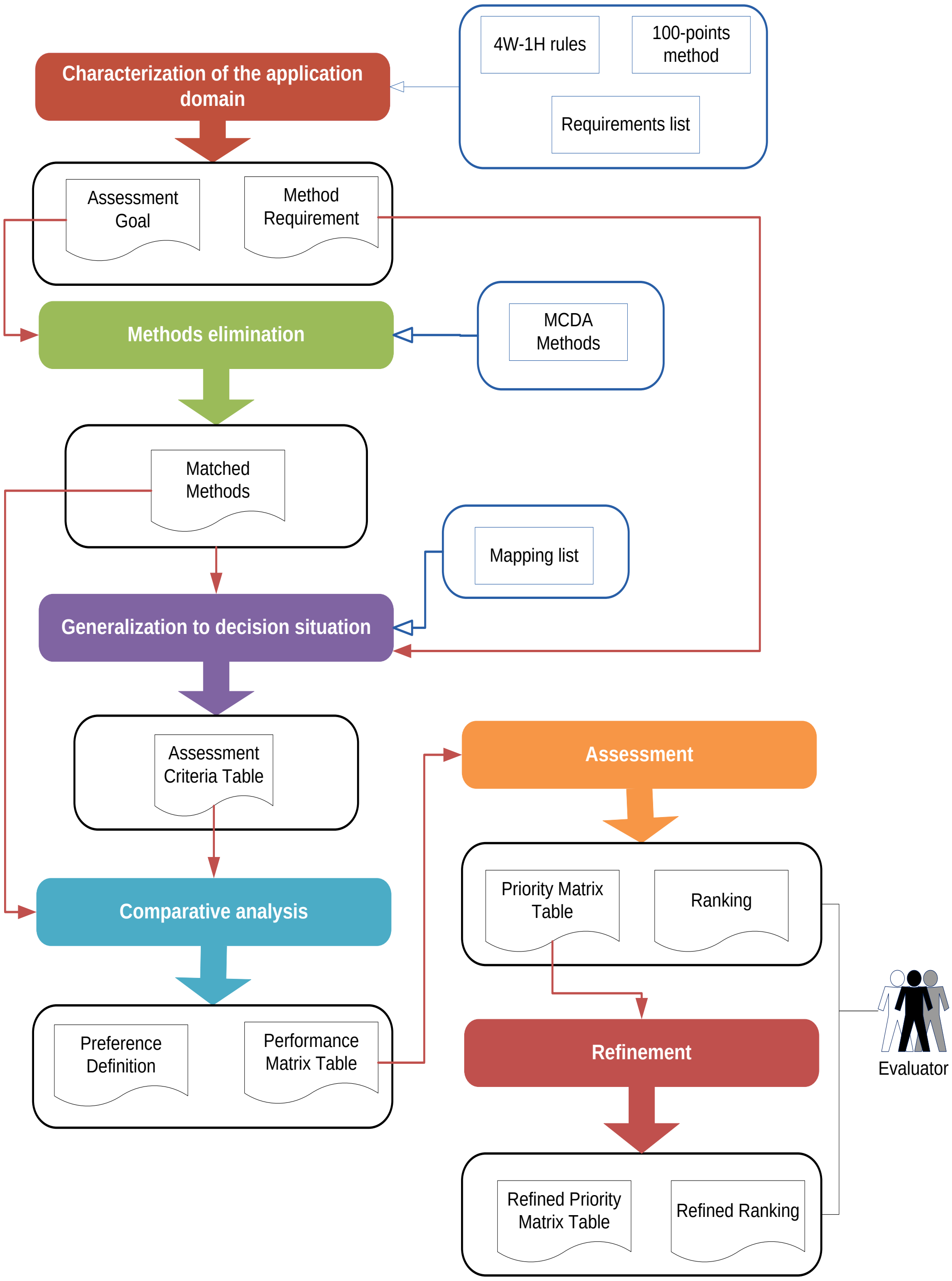

Figure 13: Overview of the MCDA-SQA Selection Framework

6.1 Characterization of the Application Domain

To select a set of suitable MCDA methods for software quality assessment (SQA), it is important to identify a set of criteria that the methods have to comply. However, the criteria to assess software quality are not always consistent because the same software product can be bound to different disciplines and regulations that can cause them to be different. As a result, the software evaluators or stakeholders have to act within a discipline or context which can influence them or be influenced by them in order to make a decision about a software product with respect to its quality assessment result. For example, the safety-critical domain requires an SQA method to have the safety attribute as a non-compensate criterion in the assessment in which the safety of the software product under study has to minimally achieve the threshold set by the regulation. However, this characteristic may not be applicable to the case for a payroll system.

Agreeing to the fact that the software quality assessment is the result of a relation between one or multiple evaluators or stakeholders influenced by a specific discipline or application domain, it becomes essential to characterize the domain where the interested software product(s) is applied. Based on the characteristic of the application domain, a set of requirements with respect to software quality assessment method can be formulated which help in selecting suitable MCDA methods in SQA.

Inspired by an approach suggested in [29], the 4W's + H rules (Why, Who, What, Which and How) is used to guide the evaluators and coordinator to characterize their application domain by considering the possible influencing factors. Figure 14 briefly illustrates how the 4W+1H rules inspire the evaluators and coordinators about the influencing factors of the application domain:

- Question "*Why to assess?*" reflects the goal to conduct the quality assessment (Section 6.1.1).
- Question *"Who to involve?"* reflects the responsible individual or group of individuals in making the decision upon the SQA result (section 6.1.2).
- Question *"What to assess?"* reflects to the set of quality criteria involves for assessment (Section 6.1.3).
- Question *"How to assess?"* reflects the possible measure used in assess the quality criteria (Section 6.1.4).
- Question *"Where to assess?"* reflects the practical issue impose on the SQA method (Section 6.1.5).

With respect to each factor, the related consequences or characteristics are further discussed. This provides a base for the evaluators and coordinator to discuss their desired characteristics of the SQA method to be applied in their domain which eventually formulates a list of their requirements about software quality assessment methods (i.e. selection requirements). Aside from specifying the SQA requirements based on 4W's + 1H rules, the framework provides a standard list of SQA requirements for the evaluators to select in order to determine their own list of selection requirements. The complete list of selection requirements is given in Table 51 (refer to Appendix G in Section 0).

After the SQA requirements have been determined, the evaluators and coordinator have to prioritize the requirements. The importance of the requirements depends on the need of it in the application domain and the preference of evaluators. Consequently, the framework applies the 100-points approach in order to reflect the different relative importance of the SQA requirements from the point of view of the evaluator(s). The evaluator is given 100-points and he has to distribute the points to all requirements defined. The evaluator gives the higher points to the requirements that are perceived more important and fewer points to the requirements that are perceived less important. If there are more than one evaluator involve, the SQA requirements are prioritized based on the normalized average priority points. Besides, the judgement from every evaluator is treated equally important. Eventually, a list of requirements together with their respective priority points is formulated.

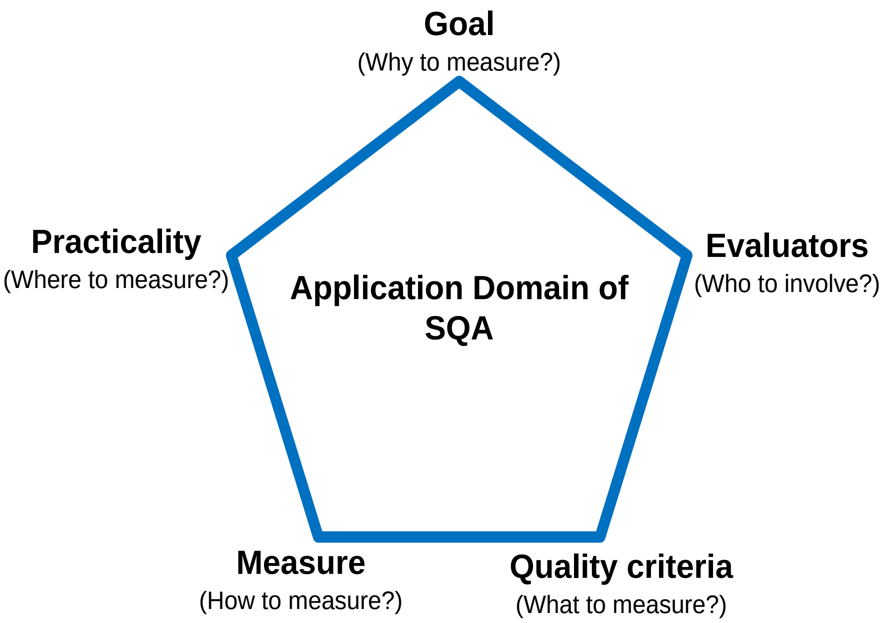

Figure 14: Influencing factors of the SQA's application domain

6.1.1 Assessment Goal

The software quality assessment requires additional resources or effort [17]. Therefore, the evaluators have a specific assessment goal to justify their decision to assess the software quality. For example, to justify whether a payroll system are worth to buy can be the goal for software quality assessment. This assessment goal also refers to the reason the coordinators are willing to spend their efforts or resources to assess software quality.

From the assessment point of view, the assessment goal affects how the assessment result should be presented. For example, to present the assessment results in a specific ranking order. There are three possible ways the coordinators can consider to present the assessment results [22, 25]:

- Rank the assessment results in a specific order.
- Group the assessment results into one of the predefined categories.
- Select a subset of the assessment results.

Essentially, the coordinators have to select one of the three types of presentation with respect to the goal they intend to achieve. This is because the way the assessment result is presented affects the way the coordinator interpret and justify whether the goal is achieved based on the result.

6.1.2 Evaluator(s)

As illustrated in Figure 14, one of the influencing factors to the application domain is the evaluators who are involved in the assessment. The evaluators here refer to the stakeholders who transform and influence their expectations to the software quality assessment. For example, the end-users, developers, the managers, etc. The result of a software quality assessment acts as a reference for evaluators to decide whether their ultimate goals have been achieved. The expectation from the evaluators about the way to assess the software quality leads to the formulation of quality criteria involved and their respective priorities.

Therefore, the second question the framework suggest is *"who to involve?"* in order to identify the possible evaluator(s) whose point of view are important in the quality assessment of software product(s) under study. By doing so, the characteristic of the application domain where the SQA take place can be determined to be either a group evaluation environment or individual evaluation environment.

From the determined evaluation environment, the coordinator can determine the evaluators' requirement whether the SQA method should be capable to facilitate the assessment under single evaluator or a group of evaluators. Further to the group evaluation environment, it is important to determine whether all stakeholders should be treated equally important or different priority. This is due to the different influencing power the evaluators possess toward the ultimate interpretation about the software quality. Besides, the coordinator has to determine whether the SQA method should be able to aggregate different preference of every evaluator about the quality criteria involved.

6.1.3 Quality Criteria

The third influencing factor of the application domain of SQA is the set of quality criteria involved (i.e. what to assess?) required in assessing the software quality. The quality criteria here refer to all the quality aspects and their respective quality attributes involve in the assessment. As illustrated in Figure 15, the coordinator has to consider five different aspects of the quality criteria that can affect the application of the software quality assessment method.

Firstly, the way the software quality is modelled in the application domain can the structure of the quality criteria the SQA method should support. For example, the quality model ISO 9126 requires a hierarchical structure of the quality criteria. Based on the quality model, the coordinator has to determine whether the method has to support certain hierarchical structure of the quality criteria. Secondly, the coordinator has to consider whether the relationship between the quality criteria is independent or interdependent while modelling the software quality. Thirdly, the coordinator has to consider whether all quality criteria have been well-defined because the uncertain definition of quality criteria requires the SQA method take fuzziness into assessment.

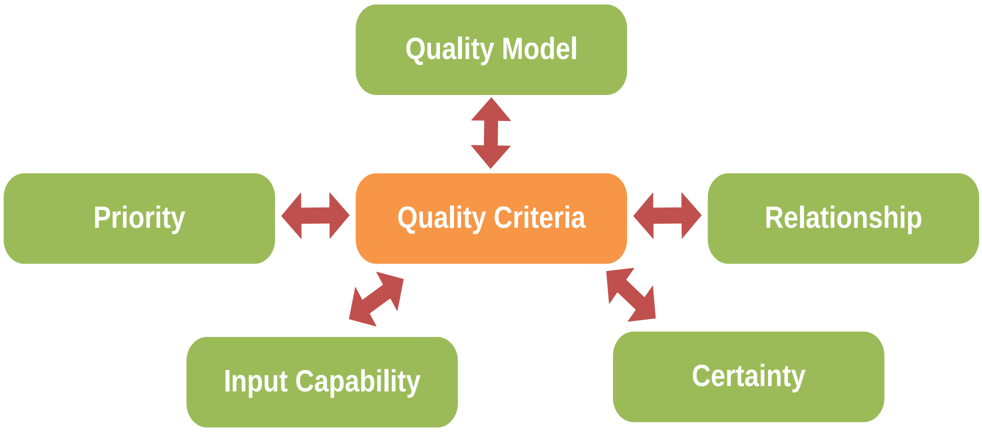

Figure 15: Different views of the quality criteria

Fourthly, the coordinator has to study the input capability of the quality criteria. The input capability of quality criteria concerns their data types, their measure scale and their threshold configuration. From all the possible input data of the quality criteria, the coordinator has to determine whether the SQA is considering the quantitative data, qualitative data or a mixture of quantitative and qualitative data. Relatively, their measure scales applied and the need of threshold value for the quality criteria can affect the application of the SQA method. Lastly, the coordinator has to decide whether all quality criteria are treated equally important or they have different priority in the assessment. This can help to identify and justify the need of SQA to support prioritization of the quality criteria.

6.1.4 Measure

To characterize the application area of SQA, the third influencing factor is the measure for the quality criteria. The result collected from the measure helps the evaluators to justify the performance of the quality criteria against their expectation. They can either define their own measures (or metrics) or select the available measure from the quality model they use (e.g. ISO 9126). However, it is not the scope of this thesis to discuss how to determine the appropriate measure for a given set of quality criteria and an assessment goal. The selection framework assumes that the measures are already determined. There are two questions the framework concerns where the first one is what kind of measures are expected to be use in SQA and how SQA method can combine these measures to represent the ultimate quality of the software.

Based on the expected measures in SQA, the evaluators or coordinator have to determine whether the measures are represented in single measure scale or multiple measure scale. This can lead to derive another requirement about the software quality assessment method. It is common in the software engineering that the information for measure may not be ready at the time of assessment. For example, the reliability attribute always relies on the prediction model to justify the measure. Therefore the measure result is imprecise and contains some degree of uncertainty. Consequently, the evaluators have to determine whether the assessment method require the measured result to be deterministic or can handle imprecise measure as the result of incomplete information.

Apart from that, the compensation degree between the measures of quality criteria required in the assessment can vary in different software engineering disciplines. Some software engineering disciplines (e.g. safety critical domain) refuse compensation for some quality criteria which means their measures results have to minimally achieve the threshold value set by the rules and regulations disregard the performance in other criteria. For example, safety-critical software (e.g. medical industry) is not allowed to compensate its poor performance in safety criterion with its better performance with respect to the other quality criteria. This is because unsafe software in safety-critical domain can cause some lost in the resources or even the worst case the loss of human life. However, the compensation degree is not fixed and relies on the domain the software applies. Therefore, the evaluators have to discuss whether compensation is allowed between the quality criteria and to which extent (refer to Section 5.1.3 for the meaning of compensation degree).

6.1.5 Practicality Issue

Lastly, the context of the software quality assessment helps the users to consider the practical consideration about the MCDA methods. The practical consideration here refers to the usage perspective about the MCDA methods. For example, the users may derive small application effort as their selection criteria based on their experience and industrial context.

The automation of the process to assess the software quality can be one of the concerns raised from the practicality issue. With the help of a software tool, the evaluators don't need to fully understand the computation as the computation can be automated from the software tool. The past experience in using any MCDA methods can also play an important role in the selection. If the evaluators have a good experience about one of the MCDA methods, they will prefer it rather than other unknown MCDA methods. Therefore, it is important to include the preference of the evaluators about the MCDA methods into consideration.

In software engineering disciplines, the reusability of the assessment can be one of the important practical issues. It is because the reusable assessment results can help to reduce the efforts spend in the assessment. Therefore, the evaluators have to determine whether they need to the SQA method to be reusable. Apart from that, some domains require the efforts or resources spend in the assessment to be minimal in order to utilize the resources in other business purposes.

6.2 Methods Elimination

The second step of the MCDA-SQA selection framework is the elimination of the non-related methods. The question is to define the meaning of non-relatedness to screen through the set of MCDA methods on hand. In this framework, the non-relatedness of the MCDA methods refers to the performance of the MCDA methods with respect to the necessary criterion. The necessary criterion is the aspect the evaluators can't tolerate if the method can't achieve or the criteria the MCDA method must possess without any excuse. Therefore, the goal of this step is to search the MCDA methods that fail to fulfil the necessary criterion and classified them to be non-related MCDA methods. These non-related MCDA methods are eliminated upfront before moving to the next step and the ultimate output from this step is the list of MCDA methods that fulfil the specified necessary criteria.

Based on the list of requirement defined from the first step, it is vital to determine what the necessary criterion is. Although the stakeholders view the meaning of software quality differently, they share a common assessment goal that makes them willing to spend effort in assessing software quality. Therefore, in our opinion, the goal of SQA should play the role of necessary criterion because it will not be significant if the selected MCDA method can't help the stakeholders to justify whether their goal is achieved. Consequently, the necessary criterion of the selection model is referred to the interpretation of the assessment goal.

Some may argue that the nature of the software engineering should also be part of the necessary criteria. However, we believe that the capability of the MCDA methods to fulfil the nature of software engineering related criteria is eventually achievable either by the methods as they are or with their minor adjustments. For example, it is common to involve a group of stakeholders for a decision making in software engineering and therefore the MCDA methods should be able to assess and apply the group interest into SQA. TOPSIS method is a MCDA method and it is unable to elicit the group interest but literally it can be used conjunctly with another MCDA method to assess the group preference upfront. Therefore, it will risk our selection if we nominate the criteria that are related with the nature of SQA as our necessary criteria. Besides, another requirement for our choice of necessary criterion is the element of consistency of the criteria. In the same example, some software industries may only involve single person for assessment and therefore the support of group decision making can't be applicable for all cases. On the contrary, a goal of assessment is needed for all possible scenario of SQA as it is the main motivation we spend resources to perform the assessment. Therefore, we believe that it is not significant if the selected MCDA method is unable to help the stakeholders to fulfil their goal of assessment.

In short, the necessary criterion in the selection framework refers to the assessment goal. Based on the assessment goal, the evaluators have to identify the proper interpretation of the assessment result that can help them to justify whether the goal has been achieved. In the decision context, the interpretation of the assessment result is conceptually transformed to the type of *problematic*, namely sorting, ranking, selecting or describing (refer to Table 7 in Section 2.4 for the meaning of each problematic type).

Once the type of problematic has been determined, a minor comparative analysis is conducted in order to find out the MCDA methods that support the desired problematic type from the list of the MCDA methods the evaluators have on hand. The MCDA methods that fail to support the desired problematic type are eliminated for further assessment. Eventually, a list of MCDA methods that support the type of problematic matched with the assessment goal are selected (i.e. the matched methods in Figure 13). Their desired presentation of the assessment result is conceptually transformed.

6.3 Generalization to Decision Situation

The third step of the selection framework is to generalize the requirements specified by the users to the matched elements in the decision context. In the framework, each requirement the evaluators specified can be conceptually mapped to one or more related elements in the decision context and these elements are known as supplementary criteria. Supplementary criteria are the aspects of MCDA methods which relate with the requirements and they are assumed to be adaptable with minor adjustment if the methods fail to fulfil (e.g. the usage of hybrid approach) or the attributes the evaluators can tolerate if the MCDA methods fail to achieve.

The conformity of the SQA requirements has to be assessed and determined based on the performance of each MCDA method with respect to each supplementary criterion. Each requirement is related with one or more supplementary criterion (i.e. aspect of MCDA methods). It is assumed that the evaluator has the knowledge about the MCDA methods or at least there is one expert having the knowledge about MCDA methods. To search the related aspect of MCDA methods, the evaluators have to discuss about which aspect of the MCDA methods can cause the positive or negative consequence against the requirements and all elements that can affect the same requirement is grouped together by using the tree structure. The negative consequences here means the methods either do not support the requirement or behave opposing against the requirement and the positive consequence means the methods possess the capability to support the requirement as specified.

The reliance of the evaluators to search appropriate supplementary criteria for a selection requirement can require a lot of effort. Therefore, the framework provides a standard mapping of the selection requirements and their supplementary criteria. Even if the intended requirement may not be found in the list, such a mapping can help the evaluators to build their understanding about the way to map the requirements to the aspect of MCDA methods (i.e., supplementary criteria). The complete mapping of selection requirement to their respective supplementary criteria is summarized in Table 52 (refer to Appendix H in Section 0).

Once the set of supplementary criteria for each selection requirement is confirmed, it is possible that there are redundant supplementary criteria and therefore the evaluators have to discuss further in order to reorganize the related supplementary criteria or inspect their requirement again whether they are related in order to group them as one composite requirement. It is required that there is no redundant supplementary criteria with respect to all requirement the evaluators specified.

Based on the list of supplementary criteria, the evaluators have to discuss and propagate the priority point of each requirement to their subsequent supplementary criteria. The distribution of priority points relies on the preference and subjective judgements from the evaluators. This is because the relative importance of each supplementary criterion with respect to the SQA requirement is not fixed and can be varied as the result of different application context or the past experience the evaluators possess. Therefore, the evaluators have to subjectively distribute the total priority point of the requirement to its related basic elements. The total priority of each element for a requirement should be equal to the priority point of the requirement. However, this subjective judgement introduces certain level of uncertainty and inconsistency to the assessment. To reduce the level of uncertainty and inconsistency, every involved evaluator is required to distribute the priority points of each requirement for at least two times and the ultimate priority points of each supplementary criterion are referred to the average of priority points.

The propagation of the priority to their related basic elements relies on the subjective judgement from the users. Figure 16 illustrates an example of the related aspect of MCDA methods with a requirement the evaluators specify "*The method shall be comprehensible*". Based on the evaluators judgement through discussion, this requirement is conceptually mapped to three supplementary criteria (i.e. aspects of MCDA methods) namely *Ease of use*, *Preference of decision maker* and *Ease of criteria weight elicitation*. It is important to note that this assignment of the aspects to the requirement is done based on the subjective judgement from the evaluators and therefore some may argue the completeness of this Ten points are given to the requirement and the evaluators need to evaluate two times in order to justify the ultimate priority point for each supplementary criterion. In this example, the priority points of each supplementary criterion are the average points and their totals should be the same as the priority points of the requirement. The output from this step is the list of supplementary criteria with respect to each requirement and their priority points.

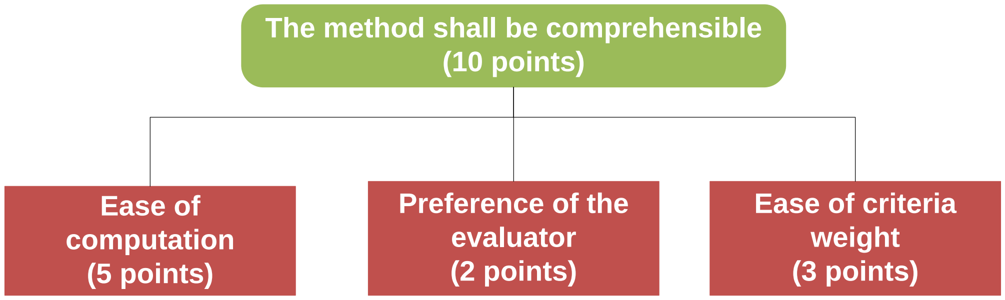

Figure 16: Example of the formulation of supplementary criteria for a SQA requirement.

6.4 Comparative Analysis

The fourth step of the selection framework is to conduct a comparative analysis on the selected method from the second step (i.e. methods elimination). The MCDA methods can have different characteristics with respect to each supplementary criterion. Before the comparative analysis is conducted, it is vital for the evaluators to define the characteristics they expect the MCDA methods to have and not to have for each supplementary criterion. The results of their preferences are documented as per the template shown in Table 31. The template covers the list of requirements and their respective supplementary criteria together with the priority points. Besides, the expectations of the evaluators about their desired or non-desired characteristics of the MCDA methods are also included.

Table 31: Template of preference table

SQA Requirement	Supplementary criteria	Criteria Weight	Desired characteristics	Non-desired characteristics

Table 32: Template of performance matrix table

Matched MCDA Method	<To fill: supplementary criterion name>	<To fill: supplementary criterion name>
<To fill: Method Name>	*<Performance of method with respect to the criterion>*	*<Performance of method with respect to the criterion>*

To assess the performance of a MCDA method with respect to all supplementary criteria, a performance matrix table has to be formulated based on the preference table defined. The template of the performance matrix table is shown in Table 32. Each row of the table is indicated as each matched MCDA method and each column illustrates the performance of the methods against each supplementary criterion. The individual performance assessment is expressed in numerical value where a value one is given if the MCDA method possesses the desired characteristic with respect to the criterion and value zero is assigned if the MCDA method shows the non-desired characteristics with respect to the supplementary criterion.

The framework treats every possible member of the desired-characteristics equally important and every member of the non-desired characteristics provide the same impact to the selection. Some may argue that the assessment result is not exact as the differences between the members that do not reflect in the assessment. Although the inclusion of the difference between the members of each category of the preference table (i.e. desired or non-desired characteristic) can lead to a more precise value, this way of assessment requires more subjective judgement from the evaluators to justify the degree of differences between the members. This has relatively added more uncertainty to the assessment and also required more efforts from the evaluators to distinguish the difference. Besides, the overall assessment process can be more complicated and less feasible. Therefore, no further refinement about the difference value for each member of the group is required and we believe that the differences between the members will not give a significant impact to the final assessment of the matched MCDA methods or the impact can be very minimal.

6.5 Assessment

The fifth step of the selection framework is to combine all the performances of the matched MCDA methods with respect to each supplementary criterion. Based on the result gathered from the performance matrix table, each performance of the method with respect to the supplementary criterion is multiplied with the criterion's priority points. The product values for each method are then added to represent the total priority points the method possesses. This addition approach is a simple and easy computation. Therefore, it won't acquire much effort from the evaluators to understand and compute. However, the framework assumes that each supplementary criterion is independent and therefore the addition approach can be applied to combine the performance value of the methods with respect to their supplementary criteria.

The degree of conformity against the SQA requirement the evaluators specify is assessed based on the sum of product value for their supplementary criteria. The higher value of a method possess shows the higher conformity of the method to suit the users in assessing the software product.

6.6 Refinement

The last step of the selection framework is to refine the priority points of the matched MCDA methods from the previous step. It aims at the non-fulfilled supplementary criteria collected from the previous steps. The evaluators have to check whether there is a solution to adapt it through the help of some software tool available or other methods. If there is a way to adapt the non-fulfil criteria in the application domain, the evaluators have to refine the priority point of the non-fulfil criteria from zero to be its priority point.

However, the introduction of certain solution to adapt the non-fulfil criterion can give certain degree of impact to the existing performance of the other criteria. Therefore, the evaluators have to analyze the impact of the solution to the existing performance of the other supplementary criteria. The framework introduces a new variable called penalty cost. The penalty cost represents the priority points in term of the degree of impact the solution give to the overall assessment by adapting the non-fulfil criteria through the help of the solution found. If the adaptation carries the negative impact, the penalty cost is given negative value and vice versa. The ultimate performance of the MCDA method is computed as the summation between the priority points and the penalty cost.

Nevertheless, it is also not the scope of this thesis to identify the way to compute the penalty cost towards the overall assessment. But the framework raises the concerns about the possible consequences to adapt the methods into certain domain. Therefore the framework requires the evaluators to assign the value of penalty cost in terms of priority point based on their experience and knowledge. The relative impact of the adaptation is worthwhile to be continued as the future work in order to provide a more reliable assessment. Besides, the framework does not require the most optimum solution for the non-fulfilled criterion but the solutions found are treated to be the same importance.

Based on the changes of the priority point for the non-fulfilled criteria as a result of its adaptation approach found, the overall performance for the related methods are modified to include its priority points and therefore a refined priority points are constructed. The changes can also affect the overall ranking of the methods and therefore a refined ranking of the matched MCDA methods have to be evaluated again.

6.7 Scenario Applying MCDA-SQA Framework

This section describes an example of scenario (Section 6.7.1) to show how our proposed MCDA-SQA framework can be applied. Based on the given example of scenario, Section 6.7.2 illustrates the way the SQA requirements are formulated and prioritized from the evaluators involve. Section 6.7.3 discusses the choice of the necessary criterion and presents the results of matched MCDA methods selected in accordance to the chosen necessary criterion. Section 6.7.4 shows how the results of the conceptually mapping between the SQA requirements the evaluators formulate to the related MCDA methods' basic element. Besides, this section also presents the results of the subjective propagation of priority points to the set of related elements. Section 6.7.5 presents the opinion of the evaluators about their desired and non-desired characteristics and constructs the performance matrix table in accordance to the comparative analysis they gathered. Based on the performance matrix table, Section 0 presents the results of priority points for each matched MCDA method and the respective ranking. Section 6.7.7 discusses the required adjustment needed to adapt MCDA method in SQA and presents the refined assessment result.

6.7.1 Example of Scenario

This scenario is constructed based on an example given in [25]. ABC corporate is a power-supply company and its current power plants are using mechanical-system controlled based reactors. However, the current reactors consume much maintenance cost and inefficient. Therefore, ABC corporate decides to purchase a software controlled protection system for the reactor in order to replace its current mechanical system.

A software vendor has offered the company their reactor controller system. The problem here is to decide whether the software can be purchased or retain the existing mechanical system. The primary concern is the new system has to be sufficiently safe which is minimally approved under the regulations. Besides, the cost and the functionalities of the new system also have to be considered. Besides the new system's maintenance cost has to be at least lower than the existing mechanical system.

ABC corporate has decided to evaluate the quality of the software prior making the decision to purchase the software. Its stakeholders, who involve in the SQA, are the company director, local community representative and the government appointed regulator. The final decision relies on the regulator as the purchased software has to be sufficiently safe against the rules and regulation.

6.7.2 Characterization of the Application Domain

Firstly, the evaluators should formulate the assessment goal and a list of related requirements based on the five influencing factors to characterize the application domain. Their goal is to ensure the proposed software controlled protection system is safe to deploy and with a reasonable price. If the system is not safe, it will not be deployed even the system is within the budget and good in other aspect. The problem here is to decide whether the proposed system is suitable for deployment that relies on the quality of the proposed system with respect to the stakeholder concerns. The evaluators involve in this scenario are the regulator, the company director and the local community representative. Each opinion of the evaluators is important to the assessment, therefore they require the SQA methods have to provide mechanism to handle their different quality goals and their respective priorities.

Based on the proposed selection framework, the quality criteria have to be determined from five areas, namely quality model, relationship, certainty, input capability and priority. From the given scenario in the previous section, the evaluators derive the software quality from three aspects they concerns namely safety, cost and functionality. The cost here refers to the maintenance cost. Furthermore, the company director requires the maintenance cost of the computerized reactor to be lesser than the existing mechanical system and the price of the proposed system to be within the company budget (i.e. ₤1.9 millions). To model this set of quality criteria, the evaluators prefer to have a hierarchical structure to represent their decomposition of the selected quality aspects as they perceive this way is easier for them to model the quality aspects and attributes they concerns. Besides, the evaluators perceive the quality criteria are interdependent and therefore they require this relations can be considered by the SQA method. According to [25], the quality criteria considered in the example are hardly to be well-defined and therefore it incurs some uncertainty. Therefore, the evaluators require the SQA method to be capable in handling the uncertainty.

The quality criteria in this scenario accept both quantitative and qualitative data type as their input. The measure scales required in this scenario are the ordinal scale and interval scale. Besides, two constraints have to be ensured in this scenario where first the price of the software has to be less than ₤1.9 millions and the safety has to be less than10^{-3} pdf. Therefore, the SQA methods have to provide the setting of threshold value for the quality criteria. Besides, each quality criterion has been treated unequally important from each evaluator. For example, the company director has more concern about the cost and the local community doesn't concern about the cost but the safety of the system. Therefore, the SQA methods have to support the quality criteria prioritization from the point of view of each evaluator. To support the prioritization, the evaluators prefer the method can provide a way for them to come out the consensually agreed priorities for the criteria (e.g. the consensus is referred to the average of the priorities collected.).

Table 33: SQA Requirement Priority Table

SQA Requirement	RG	DR	LC	Average
R1. The method shall allow different quality goal and priority from the stakeholders.	20	20	15	18
R2. The method shall support a hierarchical structure of quality modelling.	5	10	15	10
R3. The method shall handle interdependence between quality criteria.	10	5	10	8
R4. The method shall handle uncertain data.	5	5	5	5
R5. The method shall able to compute with incomplete information.	10	5	5	7
R6. The method shall support criteria weighting.	5	5	5	5
R7. The method shall support both ordinal and interval scale.	5	5	5	5
R8. The method shall support trade-off between quality criteria with restriction (i.e. partial compensation).	20	20	10	17
R9. The method shall be easily comprehensible.	10	15	20	15
R10. The method shall be supported by a software tool for reuse and automation purpose.	10	10	10	10
Sum	100	100	100	100

The evaluators look at the measures they have determined to measure the quality criteria. They found that the measures are not always deterministic because some measures provide imprecise or estimated value due to some data needed are not available at the moment. For example, the maintenance cost is measured as the estimated values that carry some impreciseness. Therefore, they prefer the SQA methods can handle uncertainty as the result of imprecise measure and assess under the context of incomplete information available at the time of assessment. Apart from that, the evaluators agree that they need to have some trade-off between the quality aspects under certain constraint. They perceive that both safety and cost aspects require partial-compensation as their trade-off are only allowed under specific constraints. For example, the local community representative wishes to compensate a safer system with higher disregard the price but the directors restrict the price have to be within the budget. Therefore, the trade-off between the cost and safety aspects is allowed only if both of them have fulfilled the minimum prerequisite (i.e. safety less than 10^{-3} pdf and the price less than ₤1.9 millions). Therefore, the SQA methods have to allow partial-compensation between the quality criteria.

All evaluators in this scenario do not have any experience in using MCDA methods and therefore the MCDA methods has to be easy for them to comprehend even they do not possess any related technical knowledge. Besides, the director prefers to have a software tool that can automate the computation of MCDA methods to assess software quality and also serve for future reuse purpose. However, the director prefers if the software tool carries low cost against the cost they need to spend in manual computation without the software tool.

In summary, ten requirements have been formulated and summarized in Table 33. Each evaluator (i.e. RG: Regulator, DR: Company Director, and LC: Local community representative) has given 100-points to prioritize 10 requirements and the results are tabulated in Table 33. The priority points of each requirement are the average of the total priority points given by all three evaluators.

6.7.3 Methods Elimination

To eliminate the non-related MCDA methods, the evaluators have to determine which type of the problematic they prefer with respect to their assessment goal. Based on the assessment goal in Section 6.7.2, their view about the aspect constitutes the appropriateness of the proposed system help to justify whether the proposed system is appropriate or not to purchase and deploy. This matches the sorting problematic where the evaluation results can be classified into two possible categories namely appropriate and inappropriate. Therefore, the evaluators have chosen the sorting problematic to be the necessary criterion.

Table 34: Minor comparative analysis result

MCDA Methods	Sorting Problematic	Remarks
AHP	Not support	
ELECTRE	Support	It is only applicable for ELECTRE TRI method.
TOPSIS	Not support	
PROMETHEE	Support	
WSM	Not Support	
VIKOR	Not support	
WPM	Not support	
ANP	Not support	
GP	Support	
UTA	Support	It is only applicable for UTA DIS method.
SMART	Support	

The evaluators have twelve core MCDA methods as their candidates namely, Analytic Hierarchical Process (AHP), Elimination and Choice Expressing the Reality (ELECTRE), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE), Weighted Sum Method (WSM), VIKOR, Weighted Product Method (WPM), Analytic Network Process (ANP), Goal Programming (GP), Utility Theory Additive (UTA), and Simple Multiple Attribute Rating

Technique (SMART) and Compromise Programming (CP). They analyze the capability of these twelve MCDA methods to support sorting problematic and the results are shown in Table 34. Five methods from the 11 candidates are capable in handling sorting problematic, namely ELECTRE TRI method, PROMETHEE family, GP method, UTADIS method, and SMART method. Therefore, these 5 methods are selected as the matched MCDA methods and the other MCDA methods candidates are eliminated for further assessment.

6.7.4 Generalization to the Decision Situation

Based on the list of basic elements given in the framework (refer to Section 11.6 for details), the evaluators have selected those basic elements of the MCDA methods that can justify the conformity of the requirements they formulated in Section 6.7.2. The related elements with respect to the requirements are the supplementary criteria and the details are summarized in Table 35. For better illustration, the overall structure of the selection requirements and their respective supplementation criteria are shown in Figure 19 (refer to Section 11.4).

From the list of supplementary criteria they selected, no redundant supplementary criterion is found. Therefore, the evaluators can proceed to distribute the priority points of the requirements to the related supplementary criteria. Based on the priority points given to each requirement, the evaluators have discussed and reached a consensus about the priority points distributed to the supplementary criterion. The details of the results are shown in Table 35.

6.7.5 Comparative Analysis

Each supplementary criteria has different characteristics, some carry binary value and some carry ordinal values. Therefore, the evaluators have to subjectively determine their desired characteristics for each supplementary criterion. All supplementary criteria S1, S5, S6, S7, S10 and S14 have only two possible characteristics, namely "*support*" or "*not support*". The desired characteristic for all these elements are decided to be "*support*" and non-desired characteristics are decided to be "*not support*".

Based on the comparative analysis table given in the framework and the preference the evaluators defined in Table 36, the evaluators assess the performance of the selected 5 methods with respect to each supplementary criterion in binary values (1 means desired characteristic and 0 means non-desired characteristics). The performances of the methods are tabulated as a performance matrix table in Table 37.

Table 35: Supplementary criteria and their priorities with respect to the requirement

SQA Requirement	Supplementary criteria	Priority Points
R1. The method shall allow different quality goal and priority from the stakeholders.	S1: Group Decision Making	5
	S2: Priority of group member inclusion	7
	S3: Fairness consideration to all stakeholders	6

R2. The method shall support a hierarchical structure of quality modelling.	S4: Organization	10
R3. The method shall handle interdependence between quality criteria.	S5: Interdependence between criteria	8
R4. The method shall handle uncertain data.	S6: Uncertainty handling	5
R5. The method shall able to compute with incomplete information.	S7: Incomplete Input information	7
R6. The method shall support criteria weighting.	S8: Support criteria weight	5
R7. The method shall support both ordinal and interval scale.	S9: Measure scale support	5
R8. The method shall support trade-off between quality criteria with restriction (i.e. partial compensation).	S10: Compensation	10
	S11: Constraint evaluation	7
R9. The method shall be easily comprehensible.	S12: Ease of use	10
	S13: Ease of criteria weight elicitation	3
	S14: Preference of the DM	2
R10. The method shall be supported by a software tool for reuse and automation purpose.	S15: Tool Assistance	5
	S16:Implementation cost	5

6.7.6 Assessment

Based on the performance result in Table 37 and the priority points of each supplementary criterion in Table 35, the evaluators construct the assessment tables where the respective priority points are given to the supplementary criteria that carry value one. The details of the assessment result are tabulated in Table 38.

For each matched MCDA method, the evaluators sum up all the priority points of its supplementary criteria. The total priority points each MCDA method possess indicate the degree of suitability of the methods to assess the software quality under the context the evaluators specified. The evaluators rank the methods based on the total priority point they possess and the ranking are illustrated in Table 38. The PROMETHEE family methods have the highest priority points and therefore they are ranked at the first place. They are followed by ELECTRE (58 points), SMART (58 points), UTA DIS (47 points) and GP (27 points). Both ELECTRE families and SMART method share the same relative percentages and therefore they are ranked the second place. Eventually, the evaluators decide to have PROMETHEE family, ELECTRE TRI method and SMART methods as their selected choices because both are statistically significant (i.e. more than half of the total priority points) and the rest are ignored as they are not statistically significant (i.e. less than half of the total priority points) from the evaluators' points of view.

6.7.7 Refinement

ELECTRE TRI method fails to facilitate group decision making, especially aggregating the group preference as the result of fairness consideration to all stakeholders. According to literature, this problem can be resolved by incorporating the MCDA methods in group decision support system (GDSS) [31]. Based on this solution, the performance of ELECTRE TRI method with respect to supplementary criterion "*S3: Fairness consideration to all stakeholders"* can be improved fully. The evaluators believe that GDSS can expedite and ease the computation but they need to spend more cost in purchasing the GDSS. Therefore, they give the impact value +5 as they perceive the impact of having GDSS is more beneficial than its negative impact. There is no other solution found to resolve the remaining weakness of ELECTRE TRI method and therefore its final refined relative percentage is 74% that rank it the second place.

PROMETHEE family methods fails to fulfil the supplementary criteria S3: Fairness consideration to all stakeholders, S4: Organization, S5: Interdependence between the criteria, S13: Ease of criteria weight elicitation and S16 Implementation cost. To adapt the PROMETHEE family methods in fulfilling the criteria S3, S4, and S5, the evaluators have found that PROMETHEE family methods can be applied conjunctly with the AHP method as suggested in [30]. Therefore, the performance of PROMETHEE family methods with respect to criteria S3, S4, and S5 can be changed accordingly as shown in. However, the evaluators found that this hybrid approach requires more efforts in the analysis and complicates the computation. There is no solution available to reduce the implementation cost. The suggested hybrid approach can increase the original implementation cost and therefore the evaluators subjectively rate the impact value as -15 to this refinement. Overall, the hybrid approach achieves the first rank with 80 points.

SMART method is not capable in handling supplementary criterion S6: Uncertainty handling and yet the literatures in [34] have suggested incorporating Fuzzy theory to overcome this shortcoming for the MCDA methods. The evaluators believe this approach help to improve the SMART methods in handling uncertainty fully and therefore they refine its performance to the full priority point of criterion S6. However, the evaluators find that the fuzzy theory is complicated to understand and therefore they need longer time to comprehend and perform the computation. This has caused them to rate the impact of incorporating fuzzy theory to be -10 and eventually produce the refined priority points of SMART method to be 53 points. No solution is found from the literature to resolve the weakness of UTA DIS and therefore its priority points remain the same. GP method has problem in handling the group decision making. As suggested in literature [31], this problem can be resolve by using the MCDA methods in group decision support system (GDSS). With the help of GDSS, GP can achieve criteria S1: Group Decision Making, S2: Priority of group member inclusion and S3: Fairness consideration to all stakeholders. Therefore, the performances of GP with respect to these three criteria are refined accordingly. The impact value of GDSS is positive to GP methods and therefore the evaluators rate it as +5. Overall, the GP method achieves a higher priority points (i.e. 50 points) and the fourth rank.

Table 36: Preference table for the example scenario

SQA Requirement	Supplementary criteria	Criteria Weight	Desired characteristics	Non-desired characteristics
R1.	S1: Group Decision Making	5	Support	Not support
	S2: Priority of group member inclusion	7	Very high, high	Low, Medium, Not support
	S3: Fairness consideration to all stakeholders	6	High, Medium	Low, Not support
R2.	S4: Organization	10	Tree, Network	Not support
R3.	S5: Interdependence between criteria	8	Support	Not support
R4.	S6: Uncertainty handling	5	Support	Not support
R5.	S7: Incomplete Input information	7	Support	Not support
R6.	S8: Support criteria weight	5	Support	Not support
R7.	S9: Measure scale support	5	Ordinal or Interval scale	Support other type of measure scale
R8.	S10: Compensation	10	Partial-compensation	Total-compensatatory, Non-compensatatory
	S11: Constraint evaluation	7	Support	Not support
R9.	S12: Ease of use	10	Very high, high	Low, Medium, Not support
	S13: Ease of criteria weight elicitation	3	Very high, high, medium	Low
	S14: Preference of the DM	2	High, Medium	Low, Not support
R10.	S15: Tool Assistance	5	Support	Not support
	S16:Implementation cost	5	Low, Medium	High

Table 37: Performance matrix table for the example scenario

MCDA Method	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16
ELECTRE TRI	0	1	0	0	0	1	1	1	1	1	1	0	0	1	1	1
PROMETHEE	1	1	0	0	0	1	1	1	1	1	1	1	0	1	1	0
SMART	1	1	1	0	0	0	0	1	0	1	0	0	1	1	1	1
GP	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	1
UTA DIS	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	1

Table 38: Assessment table for the example scenario

MCDA Method	S1	S2	S3	S4	S5	S6	S7	S8	S9	S 10	S 11	S 12	S 13	S 14	S 15	S 16	Sum	%	Rank
ELECTRE TRI	0	7	0	0	0	5	7	5	5	10	7	0	0	2	5	5	58	0.58	2
PROMETHEE	5	7	0	0	0	5	7	5	5	10	7	10	0	2	5	0	68	0.68	1
SMART	5	7	6	0	0	0	0	5	0	10	0	10	3	2	5	5	58	0.58	2
GP	0	0	0	0	0	0	0	5	0	10	7	0	0	0	0	5	27	0.27	4
UTA DIS	5	7	6	0	0	0	0	0	0	10	7	0	3	2	5	5	50	0.50	3

Table 39: Refined assessment table for the example scenario

MCDA Method	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	IV	T	R
ELECTRE TRI and AHP method	5	7	6	0	0	5	7	5	5	10	7	0	0	2	5	5	+5	74	2
PROMETHEE & AHP method	5	7	6	10	8	5	7	5	5	10	7	10	3	2	5	0	-15	80	1
SMART	5	7	6	0	0	5	0	5	0	10	0	10	3	2	5	5	-10	53	3
GP	5	7	6	0	0	0	0	5	0	10	7	0	0	0	0	5	+5	50	4
UTA DIS	5	7	6	0	0	0	0	0	0	10	7	0	3	2	5	5	0	50	4

7 PROPOSED MCDA METHODS

The aim of this chapter is to apply the proposed selection framework to select the applicable MCDA methods for software quality assessment. To propose the applicable MCDA methods, we utilize the result from both reviews that constitute the main inputs of the selection framework. The results collect from the first review serve as the input to

- Eliminate the non-related methods,
- Generalize the requirement to decision situation
- Provide the comparative analysis about the matched MCDA methods.

The collected SQA requirements from the second review are used as the input to characterize the application domain of the SQA. The following sections of this chapter illustrate the results collected from six main steps of the selection framework which constitutes the respective input information used, the intermediate result and the output product(s).

Section 7.1 illustrates a list of SQA requirements collected from second review and their respective priority points. Section 7.2 discusses the choice of the necessary criterion and presents the list of matched MCDA methods with respect to our choice of necessary criterion. Section 7.3 conceptually maps the requirements to the related MCDA methods' basic element and propagates the priority points to each related basic element. Section 7.4 presents our opinions about the desired and non-desired characteristics and transform the comparative study collected from the first review to a performance matrix table. Section 7.5 presents the computation results based on the performance matrix table and the aggregated performance for each matched MCDA method. Section 7.6 discusses the required adjustment needed to apply MCDA method in SQA and presents the refined assessment result.

7.1 Characterization of the Application Domain

The first step of the framework aims to formulate a list of selection requirements (i.e. relevant SQA requirements). It is one of our research goals to collect a list of SQA requirements in second review (i.e. Section 5.1.4). The selection requirements here are the list of SQA requirements that can characterize the application of MCDA methods in software engineering disciplines and relate with the capability of MCDA methods. The SQA requirements *Q9: The method shall have repository to store evaluation scores for future reuse purpose* and *Q21: The method shall support software improvement* are found not relevant to the characteristics of the MCDA methods and therefore they are not included as the selection requirement. Apart from that, all core MCDA methods fulfil the requirement *Q11: The method shall provide a common benchmark system* and therefore requirement *Q11* is not included as the selection requirement. Eventually, we propose selection of seventeen requirements:

- **Q1**. The method shall be comprehensable.
- **Q2**. The method shall be able to aggregate different quality attributes.
- **Q3**. The method shall support group decision making.
- **Q5**.The method shall include the consideration about the values and consequences of applying the software product.
- **Q6**.The method shall be able to perform evaluation even the information required is not complete.
- **Q8**. The method shall provide an unambigious computation procedure which all decision makers should understand it in the same way.
- **Q10** The method shall require less application effort.

- **Q12**. The method shall facilitate quality trade-offs.
- **Q13**. The method shall be repeatable.
- **Q14.** The method shall be able to handle uncertain data.
- **Q15**. The method shall be able to handle interdependencies between the quality attributes
- **Q16**. The method shall have been implemented in a software tool available to assist or automate its manual computation.
- **Q17**. The method shall express the reliability of the stakeholders' opinions in quantifiable and explicit manner.
- **Q18**. The method shall express the variance of the judgements from the stakeholders in explicit and quantifiable manner.
- **Q19**. The method should be applicable for the standard quality model. .
- **Q20**. The method shall provide reliable assessment result.
- **Q24**. The method shall facilitate the learning mechanism.

To distribute the priority points to each selection requirement based on 100-points, the numbers of votes each selection requirement possess are determined. The vote here refers to the number of publications and the number of company concerning the requirements. Figure 17 illustrates the details of 100-points distribution among these seventeen requirements.

7.2 Methods Elimination

The candidate MCDA methods here refer to the list of core MCDA methods we collected in the first systematic review (refer to Section 5.1.1 for the details). However, the assessment goal and the preferred interpretation of the evaluation results are not constant in the software engineering domain as a result of different demands and constraints of individual software disciplines or preferences of the evaluators. Therefore, our choice of the necessary criteria cannot be determined in order to eliminate the non-related MCDA methods candidates upfront. Nevertheless, each possible necessary criterion is used to group the assessment results later and the top three ranking MCDA methods in each group are analyzed. The methods with the most frequent occurring in all 3 groups of necessary criteria are selected which will be discussed in the later section. The list of matched MCDA methods from this step are the complete list of twelve core MCDA methods we found in the systematic review (refer to Section 5.1.1).

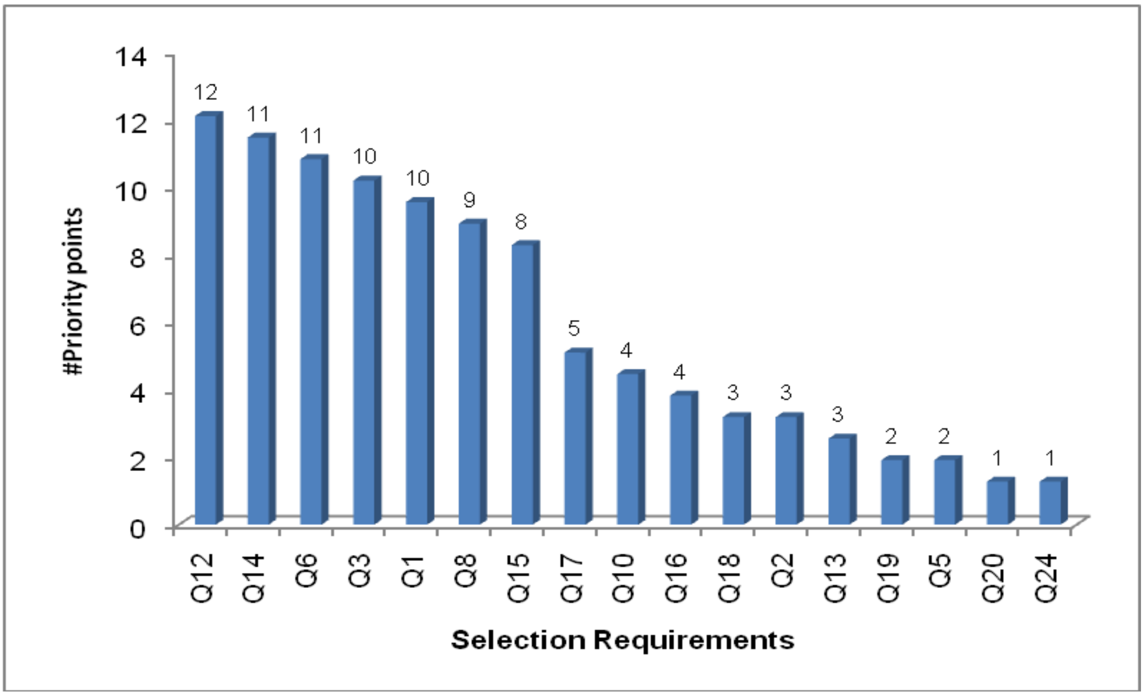

Figure 17: 100-points distribution of selection requirements

7.3 Generalization to the Decision Situation

To justify the conformity of each selection requirement specified in Section 7.1, the related basic elements of MCDA methods have to be determined which constitutes the supplementary criteria of the framework. To distribute the priority points of each selection requirement to its supplementary criteria, we treat each supplementary criteria (if there is any) of the selection requirement to be equally important and carries the average of the priority points. This is because there is no any related empirical result found to support us in judging which supplementary criterion has to be given higher priority with respect to a selection requirement.

In our opinion, the comprehensibility of the MCDA methods (i.e. *Q1*) means the degree of easiness of the methods for the evaluators to understand. Therefore, the selection requirement *Q1* relies on the easiness of the computation (i.e. criterion U1) and the simplicity of the weight elicitation approach of the method (i.e. criterion *C10*). Besides, the evaluators can easily understand or apply the MCDA methods if they have better prior experience in using the method (i.e. criterion *U17*). Therefore, in our opinion, the selection requirement Q1 has three supplementary criteria, namely *U1: Ease of use*, *U17: Preference of decision maker* and *C10: Ease of criteria weight elicitation*. As shown in Figure 17, the requirement *Q1* has a total of 9 priority points and they are equally distributed to its supplementary criterion. It is important to note that this mapping of selection requirement to the aspects of MCDA methods is done based on the subjective interpretation of this requirement Q1 with respect to the systematic review results.

Apart from that, each selection requirements *Q2, Q6, Q8, Q10, Q12, Q14, Q15, Q17, Q18, Q19, Q20,* and *Q24* have relied on a single related basic element of MCDA methods. Therefore, their supplementary criterion carries their total priority points respectively.

To justify whether a MCDA methods support group decision making (i.e. *Q3*), the capability of the MCDA method to facilitate the group decision process (i.e. *U4: Group Decision Making*) and its capability to handle the conflicting judgement from a group of evaluators (i.e. *U6: Judgement Conflict*) have to be determined. Apart from that, a MCDA method can be repeatable if it is robust to the change of the criteria and/or alternatives. Consequently, the selection requirement *Q13* is related with two elements namely *C10: Effort in criteria change* and *A4: Alternative change*. These two elements are referred as the supplementary criteria of *Q13* and therefore their priority points are the average of the total priority points of their related selection requirement. Besides, the conformity of the selection requirement Q16 relies on the existence of a software tool and also the possible implementation cost. This is because the evaluators may not prefer the methods if require higher implementation cost even the MCDA method has been automated in a software tool. Therefore, *Q16* has two supplementary criteria namely *U7: Tool assistance* and *U16: Implementation cost*.

Table 40: The supplementary criteria and their priorities WRT selection requirement

Selection Requirement	Supplementary criteria	Priority Points
Q1. The method shall be comprehensable.	U1 : Ease of use	3
	U17 : Preference of decision maker	3
	C10 : Ease of criteria weight elicitation	3
Q2. The method shall be able to aggregate different quality attributes.	P1 : Aggregation	3
Q3. The method shall support group decision making.	U4 : Group Decision Making	5
	U6 : Judgement Conflict	5
Q5. The method shall include the consideration about the values and consequences of applying the software product.	C9 : Dimension of criteria	1
	P9 : Constraint evaluation	1
Q6. The method shall be able to perform evaluation even the information required is not complete.	U18 : Incomplete input information	10
Q8. The method shall provide an unambigious computation procedure which all decision makers should understand it in the same way.	P15 : Ambiguity	8
Q10. The method shall require short application effort.	U5 : Effort spend in decision analysis	4
Q12. The method shall facilitate quality trade-offs.	P7 : Compensation	12
Q13. The method shall be reusable or repeatable.	C10 : Effort in criteria change	1
	A4 : Alternative change	1
Q14. The method shall be able to handle uncertain data.	U2 : Uncertain handling	11
Q15. The method shall be able to handle interdependencies between the quality	C5 : Interdependence between criteria	8

attributes		
Q16. The method shall have been implemented in a software tool to assist or automate its manual computation.	U7 : Tool assistance	2
	U16 : Implementation cost	2
Q17. The method shall express the reliability of the stakeholders' opinions in quantifiable and explicit manner.	U14 : Priority of group member inclusion	5
Q18. The method shall express the variance of the judgements from the stakeholders in explicit and quantifiable manner.	U15 : Fairness consideration to all stakeholder	3
Q19. The method should be applicable for the standard quality model.	U3 : Organization	2
Q20. The method shall provide reliable assessment result.	P14 : Validity	1
Q24. The method shall facilitate the learning mechanism.	U11 : Learning	1

Table 40 summarizes the details of the selections requirements with their related supplementary criteria and the respective priority points assigned. This list of supplementary criteria and the priority points will be used in the next step that is discussed in the next section.

7.4 Comparative Analysis

Based on the comparative analysis result collected from the systematic review in Section 5.1.3, each supplementary criterion (i.e. basic element of core MCDA methods) has different characteristics where some criterion can carry binary value and some carry multiple ordinal values. Besides, the supplementary criteria *P1*, *P9*, *U2*, *U3*, *U7*, *U4*, *U6*, *U18*, and *C5* have only two possible characteristics, namely "*support*" or "*not support*". The desired characteristic for all these elements is "*support*" and non-desired characteristic is "*not support*". However, the desired characteristics for each supplementary criterion with multiple ordinal values are subjectively determined based on our experience in SQA.

Based on the comparative analysis result in Section 5.1.3 and the preference information defined in Table 41, the performances of the core MCDA methods with respect to each supplementary criterion are determined and expressed in the binary values where the value one indicates its fulfilment of the desired characteristic or value zero indicates its fulfilment of the non-desired characteristics. The performance details of each core MCDA method in Table 42.

7.5 Assessment

If the supplementary criterion carries value one (refer to Table 42), its assessment result is given as its total priority points which constitutes an assessment table as shown in Table 43. All the priority points of each MCDA method with respect to its supplementary criteria are summed together as the total priority points the method achieves (i.e. column *T* in Table 43). The total priority points here indicate the degree of suitability of the methods to assess the software quality.

As discussed in Section 7.2, each type of necessary criterion is used to group the assessment results and then the each core MCDA methods in each group are ranked. The selected related MCDA methods are the core MCDA methods that fall within the top three 3 ranking and occur the most frequent in these three types of necessary criterion. The assessment results are grouped into three categories namely ranking problematic, sorting problematic and selecting problematic.

Firstly, the group of methods that support sorting problematic consists of five core MCDA methods, namely PROMETHEE family methods, SMART method, ELECTRE TRI method, GP method and UTA DIS method. Its top three ranking methods are PROMETHEE method (62 points, 1st rank), SMART method (58 points, 2nd rank) and ELECTRE TRI method (50 points, 3rd rank). The details of the MCDA methods supporting sorting problematic are illustrated in Figure 22 (refer to Section 11.5). Secondly, the group of methods that support ranking consist of all twelve core MCDA methods. The top three MCDA methods in the group of ranking problematic are AHP methods (66 points, 1st rank), PROMETHEE method (62 points, 2nd rank), SMART method (58 points, 3rd rank) and ANP method (58 points, 3rd rank). The details of the MCDA methods supporting ranking problematic are illustrated in Figure 20 (refer to Section 11.5).Thirdly, the group of methods supporting the selecting problematic consists of ten cores MCDA methods where PROMETHEE is the only method fail to support it. Its top three MCDA methods are AHP method (66 points, 1st rank), ANP method (58 points, 2nd rank), SMART method (58 points, 2nd rank) and ELECTRE method (50 points, 3rd rank). The details of the MCDA methods supporting selecting problematic are illustrated in Figure 21 (refer to Section 11.5).

Figure 18 illustrates the distribution of top three core MCDA methods in each problematic category. Six MCDA methods are found namely, AHP method, ANP method, PROMETHEE family method, SMART method, ELECTRE TRI, and ELECTRE family method. SMART is the only one occurs in three problematic groups. AHP, ANP, and PROMETHEE family methods occur in two of the problematic type. ELECTRE family (without ELECTRE TRI method) methods and ELECTRE TRI method occur only in selecting problematic and sorting problematic respectively. By considering their occurrence in these three problematic types, both ELECTRE family (without ELECTRE TRI) methods and ELECTRE TRI method have the lowest occurrence and therefore they are not strongly preferred as our proposed MCDA methods in SQA.

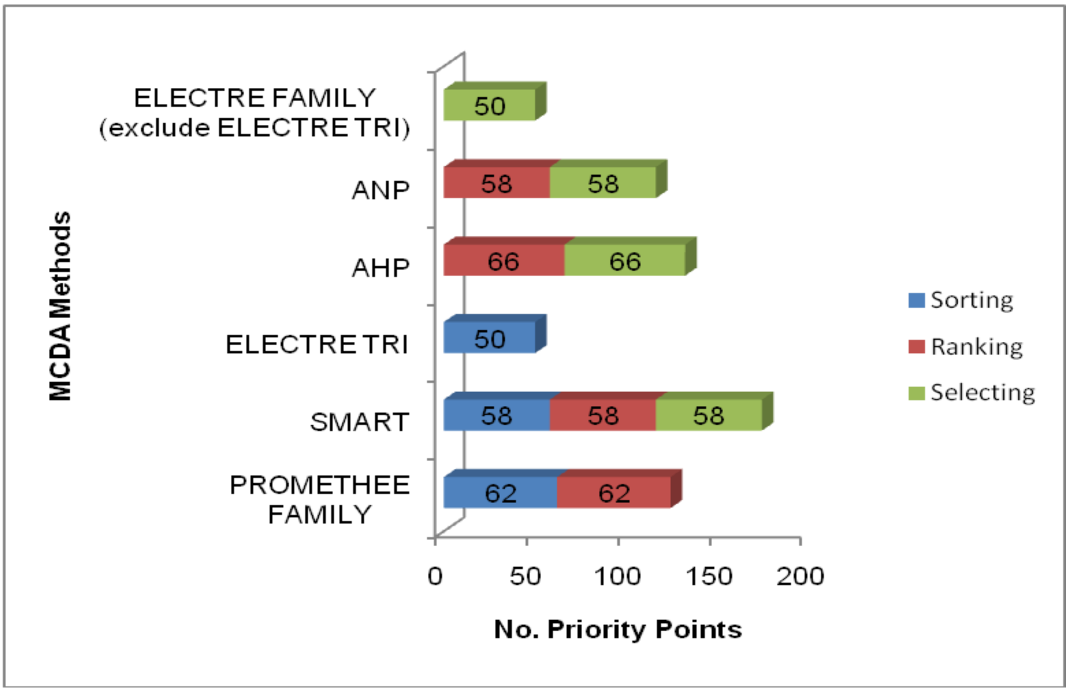

Figure 18: Distribution of top three core MCDA methods in each problematic type

7.6 Refinement of Selected Method in SQA

PROMETHEE fails to satisfy the supplementary criteria U1: Ease of use, U3: Organization, U5: Efforts spends in decision analysis, U6: Judgement Conflict, U15: Fairness consideration to all stakeholders, U16 Implementation cost, C5: Interdependence between the criteria, C9: Dimension of criteria, C10: Ease of criteria weight elicitation, and C10 : Effort in criteria change. To adapt the PROMETHEE in fulfilling the criteria U3, U6, U15, C5, and C10, the literature [30] suggests the PROMETHEE methods can be applied conjunctly with the AHP method. Therefore, the performance of PROMETHEE with respect to criteria U3, U6, U15, C5, and C10, are refined accordingly as shown in. However, this hybrid approach requires more efforts in the analysis and complicates the computation. Consequently, the original implementation cost can be relatively increased and therefore we subjectively rate the impact value to be -15 priority points for this refinement. No solution is found from the literature to resolve the remaining weakness of PROMETHEE and therefore the performance of the remaining non-fulfilled criteria are remains the same. The hybrid PROMETHEE-AHP method achieves at higher priority points (i.e. 68 points) that rank it at first place.

The SMART method, ANP method and AHP method are not capable in handling supplementary criterion U6: Uncertainty handling. A huge number of literatures have suggested applying Fuzzy theory to overcome this shortcoming for MCDA methods [34]. Therefore, the performances of these three methods with respect to criterion U2 are refined accordingly. However, some literatures criticises the application of fuzzy theory can complicate the computation of MCDA method and prolong the time for the evaluators to comprehend and perform the computation. Therefore, the impact of incorporating fuzzy theory to these three methods give more negative impact against the benefit and therefore we subjectively rate their impact value to be -10. Eventually the refined total priority points for the AHP method, SMART method and ANP method are 67 points, 59 points and 59 points respectively.

Table 41: Proposed preference construction for each proposed selection requirement.

Selection Requirement	Supplementary criteria	Priority	Desired characteristics	Non-desired characteristics
Q1	U1 : Ease of use	3	Very high, high	Medium, low
	U17 : Preference of the DM	3	High, Medium	Low
	C10 : Ease of criteria weight elicitation	3	High, Medium	Low
Q2.	P1 : Aggregation	3	Support	Not applicable
Q3.	U4 : Group Decision Making	5	Support	Not support
	U6 : Judgement Conflict	5	Support	Not support
Q5.	C9 : Dimension of criteria	1	Multiple	Single
	P9 : Constraint evaluation	1	Support	Not Support
Q6.	U18 : Incomplete input information	10	Support	Not Support
Q8.	P15 : Ambiguity	8	Non-ambiguious	Ambiguious
Q10.	U5 : Effort spend in decision analysis	4	Low	Medium, High
Q12.	P7 : Compensation	12	Partial-compensatory	Total-compensatory, non-compensatory
Q13.	C11 : Effort in criteria change	1	Low	Medium, High
	A4 : Alternative change	1	Low	Medium, High
Q14.	U2 : Uncertain handling	11	Support	Not support
Q15.	C5 : Interdependence between criteria	8	Support	Not support
Q16.	U7 : Tool assistance	2	Support	Not support
	U16 : Implementation Cost	2	Low	Medium, high

Q17.	U14 : Priority of group member inclusion	5	Very high, high, Medium	Low
Q18.	U15 : Fairness consideration to all stakeholder	3	Very high, high, Medium	Low
Q19.	U3 : Organization	2	Support	Not support
Q20.	P14 : Validity	1	High	Low, Medium
Q24.	U11 : Learning	1	Very high, high, Medium	Low

Table 42: Performance Matrix of each core MCDA methods

MCDA Method	U 1	U 2	U 3	U 4	U 5	U 6	U 7	U 11	U 14	U 15	U 16	U 17	U 18	C 5	C 8	C 9	C 10	A 4	P1	P7	P 9	P 14	P 15
AHP	1	0	1	1	0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	0	1	1
ELECTRE	0	1	0	0	0	0	1	1	1	0	0	1	1	0	1	0	0	1	1	1	1	0	0
TOPSIS	0	0	0	0	0	1	0	1	0	0	1	1	0	0	1	0	0	0	1	1	1	0	0
PROMETHEE	0	1	0	1	0	0	1	1	1	0	0	1	1	0	0	0	0	1	1	1	1	0	1
WSM	1	1	0	0	1	0	1	1	0	0	1	1	0	0	0	0	1	0	1	0	0	1	1
VIKOR	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1
WPM	1	0	0	0	1	0	1	1	0	0	1	1	0	0	0	0	1	0	1	0	0	1	1
ANP	0	0	1	1	0	1	0	1	1	1	1	0	0	1	0	1	0	0	1	1	0	1	1
GP	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	1
UTA	0	0	0	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0	1

SMART	1	0	0	1	0	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	0	1	1
CP	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Table 43: Assessment results of each core MCDA methods

MCDA Method	U1	U2	U3	U4	U5	U6	U7	U11	U14	U15	U16	U17	U18	C5	C8	C9	C10	A4	P1	P7	P9	P14	P15	T
AHP	3	0	2	5	0	5	2	1	5	3	2	3	0	8	0	3	0	0	3	12	0	1	8	66
ELECTRE	0	11	0	0	0	0	2	1	5	0	0	3	10	0	1	0	0	1	3	12	1	0	0	50
TOPSIS	0	0	0	0	0	5	0	1	0	0	2	3	0	0	1	0	0	0	3	12	1	0	0	28
PROMETHEE	0	11	0	5	0	0	2	1	5	0	0	3	10	0	0	0	0	1	3	12	1	0	8	62
WSM	3	11	0	0	4	0	2	1	0	0	2	3	0	0	0	0	1	0	3	0	0	1	8	39
VIKOR	0	11	0	0	0	5	0	1	0	0	0	0	0	0	1	0	0	0	3	12	1	0	8	42
WPM	3	0	0	0	4	0	2	1	0	0	2	3	0	0	0	0	1	0	3	0	0	1	8	28
ANP	0	0	2	5	0	5	0	1	5	3	2	0	0	8	0	3	0	0	3	12	0	1	8	58
GP	0	0	0	0	0	0	0	0	0	0	2	3	10	0	0	0	1	0	0	12	1	0	8	37
UTA	0	0	0	5	0	0	4	1	5	3	2	0	0	0	0	0	0	0	3	12	1	0	8	44
SMART	3	0	0	5	0	5	4	1	5	3	2	3	0	0	0	3	0	0	3	12	0	1	8	58
CP	0	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0	0	0	3	12	1	1	8	31

Table 44: Refined assessment results

MCDA Method	U1	U2	U3	U4	U5	U6	U7	U11	U14	U15	U16	U17	U18	C5	C8	C9	C10	A4	P1	P7	P9	P16	P17	T	IV	R
AHP	3	11	2	5	0	5	2	1	5	3	2	3	0	8	0	3	0	0	3	12	0	1	8	67	-10	2
PROMETHEE	0	11	2	5	0	5	2	1	5	3	0	3	10	8	0	3	0	1	3	12	1	0	8	68	-15	1
ANP	0	11	2	5	0	5	0	1	5	3	2	0	0	8	0	3	0	0	3	12	0	1	8	59	-10	3
SMART	3	11	0	5	0	5	4	1	5	3	2	3	0	0	0	3	0	0	3	12	0	1	8	59	-10	3

R: Rank; T: Total priority points; IV: impact value

8 THREATS TO VALIDITY

In this section, threats to the validity of this thesis work are identified and discussed. Most of the threats in this thesis work relate with the implementation of two systematic literature reviews. According to Wohlin et al. [33], the validity threats are classified into four groups, namely internal validity, external validity, conclusion validity and construct validity. The threats to validity for this thesis are discussed according to this categorization. Section 8.1 illustrates the threat to internal validity that affects the validity of the results collected from the reviews. Section 8.2 discusses the identified threats to conclusion validity that affect the results in proposed applicable MCDA methods in SQA. Section 8.3 describes the potential threats to external validity that affect the practicality of the proposed framework in real-world environment. Section 8.4 presents the threats to construct validity.

8.1 Internal Validity

In this thesis, the instrumentation threat is selected as the threat to internal validity [33]. The instrumentation threat is resulted from the bad design of the artifacts that are used in systematic reviews; in particular the selection criteria and the data extraction form. An inappropriate list of selection criteria can cause some relevant papers to be missed out. Besides, the bad design of the data extraction form can cause missing or inconsistent data extracted from the reviews that affect validity of the MCDA methods performances analysis and the grouping of similar SQA requirements. This is because more subjective judgments are expected to evaluate the MCDA when the required data is missing. Inconsistent data can cause wrong formulation of selection requirements and the corresponding selection criteria.

To mitigate this threat, the appropriateness of the selection criteria and the data extraction form has to be examined by conducting the data extraction from several researchers in parallel until a final and agreed decision is achieved. However, this approach is not possible to be conducted within the tightly-bounded schedule and regulations of this thesis. Prior to the actual publication selection, the selection criteria are reviewed by the thesis supervisor in order to seek the second opinion about its appropriateness. The data extraction form is defined in an evolutional approach where rework is expected when there is change in the extraction form. Although these two approaches are less effective than the previously mentioned cross-checking approach, it is an essential trade-off to complete this thesis within the allowed time-frame and the regulation.

8.2 Conclusion Validity

Threats to conclusion validity are the influencing factor that threats the conclusion drawn from the conducted reviews. Two threats to conclusion validity are identified in this thesis, namely reliability of measure and reliability of treatment implementation. Reliability of measure threat is caused by the extent of subjective judgment involves in the assessment of MCDA methods in the proposed framework. A more subjective assessment can lead to less reliable results because a result is considered less reliable when its assessed results may not be same for the repeated assessment on the same MCDA methods within the same context. To mitigate the subjectivity of the method assessment in the proposed framework, the performance of individual MCDA methods are assessed based on single group of criteria (i.e. desired criteria) where all the members of the group are treated equally important and the variation between the members are not included in the assessment. This can cause the method assessment result to be less precise but it is a necessary trade-off to have less subjectivity of measure and less complicated computation.

Reliability of treatment implementation threat is referred to the standard application of the proposed framework by different individuals and context. The generalization to the decision context in the proposed selection framework requires the evaluators to have MCDA methods related knowledge. Insufficient MCDA methods related knowledge can cause the evaluators to apply the procedures differently and eventually incorrect selection assessment may be expected. To mitigate the impact of this threat, the standard selection requirement list, selection criteria and comparative tables are given as the references in the framework in order to assist the evaluators to apply the framework more reliably or gain more background understanding for reliable execution of the selection procedures.

8.3 External Validity

External validity concerns about the extent the generalization of the proposed framework to different groups of individuals and software engineering disciplines [33].

The external validity to the chances to generalize the findings from systematic reviews is restricted because on one hand the systematic reviews in this thesis are conducted based on a list of predefined criteria which restrict the sample of literatures by the search date, the data sources used in the search, and the keywords in the search. On the other hand, the second reviews are consolidating the results collected from a specific number of software companies involve in the selected industrial survey. Therefore, the applicability of the findings from the systematic reviews to the whole software engineering practices can be restricted where this can affect the proposal of applicable MCDA methods to be different.

Moreover, the threats to external validity are related to the applicability of the framework in different software disciplines. However, the proposed framework is not validated in the practical environment due to the tightly bound schedule. This is important to justify whether the proposed framework is usable in the actual practical environment and therefore this threat is considerably important. A scenario study is conducted in order to analyze the practicality of the proposed model from the point of view of the author. However, some extent of the bias from the author is questionable. This threat is regarded as medium threat and therefore is worth for future work to examine the practicality of the proposed framework in the real-world setting.

Interaction of selection and treatment [33] is caused by lacking a representative sample of the population during the course of selecting applicable MCDA methods for SQA. This threat can cause the proposed applicable MCDA method(s) for SQA to be less reliable and trustable. To address this threat, the consideration about the requirements and their priority points are taken into consideration from both academia experts (i.e. from the reviews) and industrial experiences (i.e.

from the industrial survey). The industrial experts selected in the industrial survey are employed in different company with different core business.

8.4 Construct Validity

Mono-method bias, mono-operation bias and restricted generalize-ability across constructs are identified as the threats to construct validity in this thesis work [33]. Mono-method bias threat is caused by the possible prejudice from the evaluators towards one or a subset of MCDA methods base on their knowledge about single observation of the MCDA method [33]. Single observation of the MCDA method is not sufficient to explain the applicability of MCDA method in SQA. Consequently, an inappropriate MCDA method can be chosen for SQA. For example, some studies apply AHP method in SQA due to its popularity in other domains and it shows that their bias to AHP method in SQA is questionable. To avoid the measurement bias that based on single type of observation, a selection framework is proposed to include different type of observation for MCDA methods in order to cross-check the performance of MCDA methods against each other. This can help to explain the applicability of MCDA method in SQA in more comprehensive manner and lead to more suitable MCDA method to be selected for SQA.

Mono-prioritization bias threat is caused by insufficient representation of the SQA method requirements in the selection framework [33]. If the list of SQA method requirements are formulated based on the publications found in the reviews, it may not give a full picture of the needs of SQA method and cause the selection assessment in the proposed framework to be less reliable as some important selection requirements with respect to the SQA method can be missing. To mitigate this threat, the requirements about SQA method are formulated based on the consolidated results from both the review and the expert opinion in an industrial survey. By doing so, the chance of missing important SQA requirement is reduced and therefore improves the validity of the assessment of the selection framework. Yet, external validity of the survey and literature review results need to be considered.

Restricted generalize-ability across constructs is a threat concerning the assessment of improvement of applicability for MCDA method in the refinement step of the proposed selection framework (refer to Section 6.6) [33]. In this step, the identified solution for a MCDA method can affect its performance with respect to certain related criteria positively but unintentionally affect other criteria negatively. Therefore, there is a risk if the refined rating is drawn based on the improved criteria and ignoring its possible side-effect. To address this threat, the refine rating assessment is modified to consider the subjective rating value (i.e., impact value) that represents the possible side-effect of the identified improvement for the selected MCDA methods.

9 SUMMARY AND FUTURE WORK

The objective of this thesis work is twofold. On the one hand, the thesis should provide a list of MCDA methods that are appropriate for software quality assessment. On the other hand, the main issues of the suggested MCDA methods that need to be addressed for an effective software quality assessment are identified from the literatures. This objective is achieved by accomplishing four research goals as follow:

- **G1**: Identify common MCDA methods and analyze them for the purpose of identifying and understanding their basic components and characteristics.
- **G2**: Identify the criteria for the purpose of selecting MCDA methods that are most suitable for software quality assessment.
- **G3**: Define systematic procedure for selecting and suggesting the MCDA method(s) that is best suitable in accordance to the requirements of the software quality assessment.
- **G4**: Select MCDA methods candidates suitable for assessing software quality and identify their weaknesses.

The way to achieve these four research goals is summarized in Table 45. Both research goals G1 and G2 are completely addressed by conducting the systematic literature reviews which answer the research questions RQ1.1, RQ 1.2, RQ 2.1 and RQ 2.2 respectively. Besides, both research goals G3 and G4 are addressed partially as the further validation is lacking for G3 and the complete search of optimum adaptation is lacking for G4.

The benefits and issues of single MCDA base methods are studied for research goal G1, however many publications have been encountered suggesting the usage of integrated MCDA methods instead of single MCDA base method to overcome the shortcoming of the MCDA method. The integrated MCDA method here includes the hybrid MCDA methods and the concurrent usage of two or more distinct MCDA base methods. This opens the work of research goal G1 to the integrated MCDA method.

The following section presents a summary of how the initially specified research questions are addressed in the course of this thesis work while Section 9.2 suggests the possible future works.

In overall, the main contributions of this thesis work are:

- providing a comparative analysis result for the common MCDA methods
- presenting the potential requirements that the MCDA methods have to satisfy for SQA
- proposing a selection framework which assist the evaluators to select suitable MCDA methods in software quality assessment
- suggesting a list of MCDA methods that are appropriate to be used for software quality assessment

Table 45: Summary of the research goals and their corresponding implemented solution

Research Goal ID	Research Methods	Addressed by	Extent of achievement
G1	Systematic literature review	Research question RQ1.1, RQ 1.2.	Complete
G2	Systematic literature review, literature review	Research questions RQ 2.1, RQ 2.2	Complete
G3	Conceptual Analysis	Research question RQ 3.1	Partial as the proposed framework is not fully validated in the real world situation due to the consideration of the tightly bounded schedule. The validation of the proposed framework is suggested as the future work.
G4	Conceptual Analysis, literature review	Research question RQ 4.1, RQ 4.2	Partial as the complete screening of the optimum way to adapt the selected MCDA method in SQA is omitted due to the consideration of the tightly bounded schedule. Therefore, this issue is suggested to be addressed in the future work.

9.1 Research Questions Revisited

In this thesis, two systematic reviews are conducted. The first systematic review is conducted to study the characteristics of the MCDA methods available in the literature. The outcome of this review assists in answering RQ 1.1, RQ 1.2 and RQ 2.1. The second systematic review is a mini review focuses on the publications in five specific software engineering data sources, namely International Conference on Software Engineering (ICSE), International Symposium on Empirical Software Engineering and Measurement (ESEM), International Conference on Software Engineering and Knowledge Engineering (SEKE), Product Focused Software Development and Process Improvement (Profes), and Software & Systems Engineering Essentials (SEE). This review aims to answer the research question RQ 2.2.

By analyzing the results collected from both reviews, MCDA-SQA selection framework is proposed in order to guide the evaluators in selecting suitable MCDA methods to assess software product quality in which answering research question RQ 3.1. The proposed framework is instantiated by using the information from both reviews in order to propose the most suitable MCDA method(s) in software quality assessment that answering RQ 4.1. Based on the list of MCDA methods we proposed, some adjustments are suggested in order to adapt the methods in the software engineering environment which are answering research question RQ 4.2. The results of each research question are summarized in the following sections respectively.

9.1.1 RQ1.1: What are the core MCDA methods?

To answer this research question, the frequently analyzed MCDA base methods are identified as the core MCDA methods. In the review, twenty-one MCDA base methods are found and AHP method is the most frequently analyzed method. Out of the twenty-one identified MCDA base methods, twelve of them are determined as the core MCDA methods due to their occurrences exceed the minimum threshold value. In conclusion, the core MCDA methods found from this review are Analytic Hierarchical Process (AHP), Elimination and Choice Expressing the Reality (ELECTRE), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE), Weighted Sum Method (WSM), VIKOR, Weighted Product Method (WPM), Analytic Network Process (ANP), Goal Programming (GP), Utility Theory Additive (UTA), and Simple Multiple Attribute Rating Technique (SMART).

9.1.2 RQ1.2: What are basic elements common for all MCDA methods?

The publications collected in the first review compare the MCDA methods from different aspects homogeneous environment or domains they involve in. In summary, there are forty-nine basic elements are found in this review and their details are listed as follow:

- *U1: Ease of use*
- *U2: Uncertainty handling*
- *U3: Organization*
- *U4: Group Decision Making*
- *U5:Effort spend in decision analysis*
- *U6: Judgment conflict resolution*
- *U7: Tool assistance*
- *U8: Applicability to group decision making [43]*
- *U9: Dimension of decision problem*
- *U10: Leadership assurance [43]*
- *U11: Learning*
- *U12: Scope [43]*
- *U13: Assistance of alternative development [43]*
- *U14: Priority of group members inclusion*
- *U15: Fairness consideration to all stakeholders [43]*
- *U16: Implementation cost*
- *U17: Preference of decision maker*
- *U18: Incomplete input information*
- *C1: Nature of criteria*
- *C2: Optimum size of criteria.*
- *C3: Supporting criteria weight*
- *C4: Non-commensurable criteria handling*
- *C5: Interdependence between criteria*
- *C6: Nature of criteria weight in the MCDA method*
- *C7: Supported measurement scale*
- *C8: Dimension of criteria*
- *C9: Ease of criteria weight elicitation*
- *C10: Effort in criteria change*
- *A1: Optimum size of alternative*
- *A2: Nature of alternative*
- *A3: Interdependence between alternatives*
- *A4: Alternative change*
- *P1: Aggregation*
- *P2: Types of problematic*
- *P3: Preference model*

- *P4: Alternative evaluation*
- *P5:Completeness*
- *P6: Required model parameter setting*
- *P7: Compensation [40]*
- *P8: Rank reversal*
- *P9: Constraint inclusion*
- *P10: Risk evaluation*
- *P11:Support relation between decision level*
- *P12: Aggregation condition*
- *P13: Moment to include preference*
- *P14: Validity of evaluation*
- *P15: Ambiguity of evaluation result*
- *P16:Transitivity*
- *P17:"Psychophysical applicability" [43]*

To get an overview of what is actually evaluated from these basic elements, all the basic elements are categorized according to the basic concept of MCDA [26], namely the criteria, preference evaluation, the alternative and the usage. It is notable that the usage facet has the highest number of elements among the four categories and it is followed closely by the preference evaluation facet.

9.1.3 RQ2.1: What are the most relevant characteristics of the MCDA methods to be considered when selecting MCDA method for general decision making problems?

To answer RQ 2.1, all the characteristics of the MCDA method with respect to each element in the selected papers are accumulated to get an overview of which characteristics are actually being considered. From the forty-nine basic elements found in the first review, the way to measure the basic elements are identified and then the possible values of the basic elements are determined. These possible values of basic elements are designated as the characteristics of MCDA method.

To determine which characteristics are the most relevant to be considered, the number of publications their basic element is assessed. From this review, the characteristics of element *C7: Supported measurement scale* are identified as the most relevant because this element has the highest occurrence of publications (i.e. 19 publications). It is followed by the characteristics of the element *P4: Alternative evaluation* (i.e. 16 publications). The characteristics of both elements P1: Aggregation and P2: Types of problematics shares the same occurrence (i.e. 14 publications).

However, it is observed that all basic elements collected from the review have involved at least two or more MCDA methods and therefore we believe that the characteristics of all the identified basic elements can be relevant to be considered during the selection of suitable MCDA method for general decision making.

9.1.4 RQ2.2: What are the most relevant requirements regarding software quality assessment?

By referring to the motivations to introduce new SQA method and the issues about the existing SQA methods collected in second review, the requirements of the software quality assessment method are identified and formulated. To strengthen the review results, a statistical result from an industrial survey is used to consolidate with review result and formulate the list of relevant SQA requirement.

In total, twenty-four requirements regarding software quality assessment method are collected and twenty-one of them are selected as the most relevant requirements as their votes have exceeded one vote which is the minimum threshold value we set. The details of the requirements are illustrated in Table 51 (refer to Section 0). The review results show that at least half of the studies have been discussed the following three requirements in assessing software quality:

- *R11. The method shall support quality trade-off*
- *R13. The method shall be able to handle uncertain data.*
- *R06:* The method shall be able to perform evaluation even the information required is not complete.

9.1.5 RQ 3.1: How to systematically select the most suitable MCDA method based on the criteria defined in RQ 2.1 and RQ 2.2?

From the analysis of the previously answered research questions and taking into account the challenging in selecting suitable MCDA for SQA, the major aspect that need to be considered for effective evaluation of software quality are identified and a selection framework is proposed to answer research question RQ3.1. The framework consists of six main steps:

Step 1: Characterization of the application domain,
Step 2: Method elimination,
Step 3: Generalization to decision situation,
Step 4: Comparative analysis
Step 5: Assessment
Step 6: Refinement.

The overviews of these six steps are summarized in Table 46.

Table 46: Summary of steps in the selection framework

Steps	Input	Output
Characterization of the application domain	None	Selection requirements
Method elimination	Necessary criterion, MCDA method candidates	Matched MCDA methods
Generalization to the decision situation	Selection Requirements	Mapped basic elements of MCDA method
Comparative analysis	Matched MCDA methods, mapped basic elements	Preference Table, comparative matrix table
Assessment	Comparative matrix table	Performance matrix table
Refinement	Performance matrix table	Refined performance matrix table, list of adaptations

9.1.6 RQ4.1: Which existing MCDA methods are most suitable for software quality assessment?

To answer RQ4.1, the proposed selection framework is instantiated with the results collected from both reviews. Firstly, the SQA requirements from the second review are used as the selection requirements and their votes are normalized in 100-point scale to indicate their priority point. Secondly, both the basic elements and comparatives analysis of the MCDA methods are used to serve as inputs for the second, third and fourth steps. Since there are three possible interpretation results (i.e., ranking, selecting, and sorting) for SQA, the necessary criteria cannot be determined and therefore no MCDA method is eliminated upfront. The performances of MCDA methods are assessed and the results are classified into aforementioned interpretation results.

The AHP method carries 66 priority points and this makes it to be the most suitable method for the group of selecting and ranking. In the group of sorting, the PROMETHEE family carries 62 priority points and therefore it is the most suitable method. Although SMART method is not the most suitable by considering its total priority points (i.e. 58 points), it is the only method that can support all three possible types of interpretation SQA results. In conclusion, AHP method, PROMETHEE family methods and SMART method are the most suitable methods for software quality assessment. However, each of them has some weaknesses that are needed to be adapted in software quality assessment.

9.1.7 RQ 4.2: What are remaining deficits of MCDA methods identified in RQ4.1 and what are potential solutions to these deficits?

To answer RQ 4.2, a solution is searched to resolve the non-fulfilled basic elements of the suitable methods. In software engineering environment, uncertainty is a common issue because it is difficult to ensure all prerequisite information to be ready at the time of assessment. Therefore, it is preferable to have MCDA methods to support these uncertainties but both SMART and AHP methods are incapable in handling this. To adapt AHP method and SMART methods in handling uncertainty, fuzzy theory is suggested to be applied in these two methods to overcome this shortcoming [34]. However, the fuzzy logic is more complicated to comprehend and therefore this adaptation gives a negative impact to the overall performance of the methods. PROMETHEE family methods have a vast problem in facilitating group decision process which is common in software quality assessment. A literature is identified and it suggested the usage of hybrid approach of AHP-PROMETHEE to strengthen the PROMETHEE family methods in handling group decision environment [30].

9.2 Future Work

This thesis has proposed a selection framework to guide the evaluators in selecting suitable MCDA methods in SQA. There are several potential future works to follow up the results of this thesis and they include:

- **Validate the method selection framework**
  As a result of tightly bound schedule of this thesis work, MCDA-SQA selection framework is not validated by applying it on actual industrial setting or validate against the expert opinions. The validation of MCDA-SQA framework on a live software engineering environment is essential to check whether the framework is practical and usable in the actual practice and adjustable to different software engineering discipline. The potential feedback from the validation can be collected to further improve the framework.

- **Expansion of the systematic review**
  An extension of the systematic reviews involving more databases, in particular the requirements about the software quality assessment in the second review in order to collect more information. This can help to formulate a more reliable and complete list of the requirements to be incorporated in the framework.

- **Improvement of the framework**
  The proposed framework in this thesis is motivated and primarily based upon the findings from both systematic reviews. Aside from the systematic reviews, the framework can be improved by acquiring more inputs and opinions from the practitioners especially in the area of software quality assessment.

- **Investigate the adaptation of MCDA methods**
  From the first review, numerous papers discuss hybrid MCDA approaches in different application domains. Yet, we excluded them from consideration because our focus was on the basic MCDA methods. However, it will be interesting to investigate the adaptations available for each MCDA method and their impact on their suitability for SQA. This information can later on further improve the proposed framework especially in specifying the impact value to include the suggested adaptation.

- **Tool automation**
  As illustrated in the framework description (i.e., Chapter 6) and the scenario application (i.e. Chapter 7), the procedures in the proposed framework are systematic and follow arranged steps. It will be reasonable to implement a tool to incorporate the proposed framework together with the software quality assessment. This can help the practitioner to manage the artefacts related in the assessment and make the assessment to be more effective.

10 REFERENCES

1. D. Prasad and J. McDermid, "Dependability evaluation using a multi-criteria decision analysis procedure," in *Dependable Computing for Critical Applications 7, 1999*, 1999, 339-358.
2. Kilsup Lee and Sung Jong Lee, "A quantitative software quality evaluation model for the artifacts of component based development," in *6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and 1st ACIS International Workshop on Self-Assembling Wireless Networks, SNPD/SAWN 2005, May 23, 2005 - May 25, 2005*, vol. 2005, Proceedings - Sixth Int. Conf. on Softw. Eng., Artificial Intelligence, Netw. and Parallel/Distributed Computing and First ACIS Int. Workshop on Self-Assembling Wireless Netw., SNPD/SAWN 2005 (Towson, MD, United states: Institute of Electrical and Electronics Engineers Computer Society, 2005), 20-25, http://dx.doi.org/10.1109/SNPD-SAWN.2005.7.
3. T. Rosqvist, M. Koskela, and H. Harju, "Software quality evaluation based on expert judgement," *Software Quality Journal* 11, no. 1, Softw. Qual. J. (USA) (May 2003): 39-55.
4. McCall, J. A., Richards, P. K., and Walters, G. F., "Factors in Software Quality", *Nat'l Tech. Information Service,* no. Vol. 1, 2 and 3, 1977.
5. Boehm, B.W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M., Characteristics of Software Quality, North Holland, 1978.
6. Kitchenham, B. A., Charters, S., "Guidelines for performing Systematic Literature Reviews in Software Engineering", Keele University, EBSE, Technical Report, EBSE-2007-01.
7. M. Beaney, "Analysis", The Stanford Encyclopedia of Philosophy, Published April 7, 2003 [Online]. Available: The Stanford Encyclopedia of Philosophy, http://plato.stanford.edu/entries/analysis [Accessed: Apr 12, 2010]
8. A. Koscianski and J. Candido Bracarense Costa, "Combining analytical hierarchical analysis with ISO/IEC 9126 for a complete quality evaluation framework," in *Proceedings of IEEE Computer Society 4th International Symposium and Forum on Software Engineering Standards, 17-21 May 1999*, Proceedings 4th IEEE International Software Engineering Standards Symposium and Forum (ISESS'99). `Best Software Practices for the Internet Age' (Los Alamitos, CA, USA: IEEE Comput. Soc, 1999), 218-26, http://dx.doi.org/10.1109/SESS.1999.766597.
9. Che-Wei Chang, Cheng-Ru Wu, and Hung-Lung Lin, "A simplified measurement scheme for software quality," *Journal of Information & Optimization Sciences* 27, no. 3, J. Inf. Optim. Sci. (India) (September 2006): 723-32.
10. ISO, International Organization for Standardization, "ISO-9126-I:2001, Software engineering – Product quality, Part 1: Quality model", 2001.
11. ISO, International Organization for Standardization, "ISO-9126-2:2001, Software engineering – Product quality, Part 2: External metrics", 2001.
12. ISO, International Organization for Standardization, "ISO-9126-3:2001, Software engineering – Product quality, Part 3: Internal metrics", 2001.
13. ISO, International Organization for Standardization, "ISO-9126-4:2001, Software engineering – Product quality, Part 1: Quality In Use metrics", 2001.

14. Figueira, José; Salvatore Greco, Matthias Ehrgott (2005). *Multiple Criteria Decision Analysis: State of the Art Surveys*. New York,: Springer Science + Business Media, Inc.. ISBN ISBN 0-387-23081-5.
15. N. Bevan, "Quality in use: meeting user needs for quality," Journal of Systems and Software 49, no. 1, J. Syst. Softw. (USA) (December 15, 1999): 89-96.
16. David A. Garvin, "WHAT DOES 'PRODUCT QUALITY' REALLY MEAN?," Sloan management review 26, no. 1 (1984): 25-43.
17. B. Kitchenham and S.L. Pfleeger, "Software quality: the elusive target [special issues section]," IEEE Software 13, no. 1, IEEE Softw. (USA) (January 1996): 12-21.
18. Yang Aimin and Zhang Wenxiang, "Based on quantification software quality assessment method," Journal of Software 4, no. 10 (2009): 1110-1118.
19. T.M. Khoshgoftaar and Kehan Gao, "Count models for software quality estimation," IEEE Transactions on Reliability 56, no. 2, IEEE Trans. Reliab. (USA) (June 2007): 212-22.
20. M.-A. Cote, W. Suryn, and E. Georgiadou, "In search for a widely applicable and accepted software quality model for software quality engineering," Software Quality Journal 15, no. 4, Softw. Qual. J. (Netherlands) (December 2007): 401-16.
21. B. Kitchenham and J.Walker, "The Meaning of Quality," in Software Engineering 86, 10-12 Sept. 1986
22. Norman E. Fenton and Shari Lawrence Pfleeger, Software Metrics: A Rigorous & Practical Approach, 2nd ed., Boston: PWS Pub, 1997.
23. P. Vincke, Multicriteria Methodology for Decision Aiding, Kluwer Academic, Dordrecht, 1996.
24. E. Lken, "Use of multicriteria decision analysis methods for energy planning problems," Renewable and Sustainable Energy Reviews, vol. 11, 2007, pp. 1584-1595.
25. N. Fenton and M. Neil, "Making decisions: Using Bayesian nets and MCDA," Knowledge-Based Systems, vol. 14, 2001, pp. 307-325.
26. Vansnick, J.C. (1990), "Measurement theory and decision aid", in Bana e Costa (ed.), Readings in Multiple Criteria Decision Aid, Springer-Verlog, Berlin, 81-100.
27. J. Ananda and G. Herath, "A critical review of multi-criteria decision making methods with special reference to forest management and planning," Ecological Economics, vol. 68, 2009, pp. 2535-2548.
28. E. Jacquet-LagreÌze and Y. Siskos, "Preference disaggregation: 20 Years of MCDA experience," European Journal of Operational Research, vol. 130, 2001, pp. 233-245.
29. L. Buglione and A. Abran, "Assessment of Measurement Indicators in Software Process Improvement Framework," Dec. 2008.
30. J. Ignatius, S. Motlagh, M. Sepehri, M. Behzadian, and A. Mustafa, "Hybrid models in decision making under uncertainty: The case of training provider evaluation," Journal of Intelligent and Fuzzy Systems, vol. 21, 2010, pp. 147-162.
31. G. Rigopoulos, N. Karadimas, and A. Orsonni, "Modeling group decision-making for collaborative teams in enterprises," 2008 Second Asia International Conference on Modeling & Simulation, 13-15 May 2008, Piscataway, NJ, USA: IEEE, 2008, pp. 738-43.
32. N.F. Schneidewind, "Body of knowledge for software quality measurement," Computer, vol. 35, 2002, pp. 77-83+13.

33. C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in software engineering: an introduction, Kluwer Academic Publishers, 2000.
34. S. Onut, T. Efendigil, and S. Soner Kara, “A combined fuzzy MCDM approach for selecting shopping center site: An example from Istanbul, Turkey,” Expert Systems with Applications, vol. 37, 2010, pp. 1973-1980.
35. Z. Ayag and R. Ozdemir, “An intelligent approach to ERP software selection through fuzzy ANP,” International Journal of Production Research, vol. 45, 2007, pp. 2169-2194.
36. P. Juliff, “Software quality function deployment,” Proceedings of Second International Conference on Software Quality Management. SQM 94, 26-28 July 1994, Southampton, UK: Comput. Mech. Publications, 1994, pp. 533-42.
37. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-Functional Requirements in Software Engineering. International Series in Software Engineering, vol. 5, p. 476. Springer, Heidelberg (1999).
38. Xiaojing Liu and Jihong Pang, “A Fuzzy Synthetic Evaluation Method for Software Quality,” 2010 2nd International Conference on E-business and Information System Security (EBISS 2010), 22-23 May 2010, Piscataway, NJ, USA: IEEE, 2010, p. 4 pp.
39. Hoyer, R.W. and Hoyer, B.B. Y., “What is quality?,” Quality Progress, 2001, no. 7, pp. 52-62.
40. Guitouni and J. Martel, “Tentative guidelines to help choosing an appropriate MCDA method,” European Journal of Operational Research, vol. 109, 1998, pp. 501-521.
41. Y. Levy and T.J. Ellis, “A systems approach to conduct an effective literature review in support of information system research,” Informing Science: International Journal of an Emerging Transdiscipline, vol. 9, 2006, pp. 181-212.
42. Carol Busch, Paul S. De Maret, Teresa Flynn, Rachel Kellum, Sheri Le, Brad Meyers, Matt Saunders, Robert White, and Mike Palmquist. (2005). Content Analysis. Writing@CSU. Colorado State University Department of English. [Online]. Available: http://writing.colostate.edu/guides/research/content/
43. K. Peniwati, “Criteria for evaluating group decision-making methods,” Mathematical and Computer Modelling, vol. 46, 2007, pp. 935-947.
44. Dodson, J., Spackman, M, Pearman, A.D. & Phillips, L.D. , “Multi-Criteria Analysis: a Manual Department of the Environment”, 2000, Transport and the Regions, London
45. J. Aranda, N. Ernst, J. Horkoff, and S. Easterbrook, “A framework for empirical evaluation of model comprehensibility,” ICSE 2007 Workshops: International Workshop on Modeling in Software Engineering, MISE'07, May 20, 2007 - May 26, 2007, Minneapolis, MN, United states: Inst. of Elec. and Elec. Eng. Computer Society, 2007, p. ACM SIGSOFT; IEEE Computer Society.
46. Belton, S., Stewart, T.S., 2002, Multiple Criteria Decision Analysis. An integrated Approach. Kluwer Academic Publishers, Massachusetts.
47. A. Trendowicz, M. Klaes, C. Lampasona, "Quality Assessment Method", Technical Report, Quamoco Project, Kaiserslautern 2010.
48. H. Shyur, “COTS evaluation using modified TOPSIS and ANP,” Applied Mathematics and Computation (New York), vol. 177, 2006, pp. 251-259.
49. G. Buyukozkan and D. Ruan, “Evaluation of software development projects using a fuzzy multi-criteria decision approach,” Mathematics and Computers in Simulation, vol. 77, 2008, pp. 464-475.

50. A. Jadhav and R. Sonar, “Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study,” 2009 2nd International Conference on Emerging Trends in Engineering and Technology, ICETET 2009, December 16, 2009 - December 18, 2009, Nagpur, India: IEEE Computer Society, 2009, pp. 991-997.
51. L. Shi and Y. Shanlin, “The Evaluation of Software Trustworthiness with FAHP and FTOPSIS Methods,” 2009, pp. 1-5. FTOPSIS Methods,” 2009, pp. 1-5
52. V.R. Basili, G. Caldiera, and H.D. Rombach, “The goal question metric approach,” Encyclopedia of Software Engineering, vol. 1, 1994, pp. 528-532.
53. F. Akhavi and C. Hayes, “A comparison of two multi-criteria decision-making techniques,” Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2003, pp. 956-961.
54. H. Shyur and H. Shih, “A hybrid MCDM model for strategic vendor selection,” Mathematical and Computer Modelling, vol. 44, 2006, pp. 749-761.
55. G. Kiker, T. Bridges, A. Varghese, P. Seager, and I. Linkov, “Application of multicriteria decision analysis in environmental decision making.,” Integrated environmental assessment and management, vol. 1, 2005, pp. 95-108.
56. S. Pohekar and M. Ramachandran, “Application of multi-criteria decision making to sustainable energy planning - A review,” Renewable and Sustainable Energy Reviews, vol. 8, 2004, pp. 365-381.
57. P. Zhou and B. Ang, “Comparing MCDA aggregation methods in constructing composite indicators using the Shannon-Spearman measure,” Social Indicators Research, vol. 94, 2009, pp. 83-96.
58. E.W. Cheng and H. Li, “Contractor selection using the analytic network process,” Construction Management and Economics, vol. 22, 2004, pp. 1021-1032.
59. F. Torfi, R.Z. Farahani, and S. Rezapour, “Fuzzy AHP to determine the relative weights of evaluation criteria and Fuzzy TOPSIS to rank the alternatives,” Applied Soft Computing Journal, vol. 10, 2010, pp. 520-528.
60. J. Malczewski, “GIS-based multicriteria decision analysis: a survey of the literature,” International Journal of Geographical Information Science, vol. 20, 2006, pp. 703-26.
61. A. Moffett and S. Sarkar, “Incorporating multiple criteria into the design of conservation area networks: A minireview with recommendations,” Diversity and Distributions, vol. 12, 2006, pp. 125-137.
62. L. Diaz-Balteiro and C. Romero, “Making forestry decisions with multiple criteria: A review and an assessment,” Forest Ecology and Management, vol. 255, 2008, pp. 3222-3241.
63. N.F. Matsatsinis and A.P. Samaras, “MCDA and preference disaggregation in Group Decision Support Systems,” European Journal of Operational Research, vol. 130, 2001, pp. 414-429.
64. G. Mendoza and H. Martins, “Multi-criteria decision analysis in natural resource management: A critical review of methods and new modelling paradigms,” Forest Ecology and Management, vol. 230, 2006, pp. 1-22.

65. I. Linkov, F. Satterstrom, G. Kiker, T. Seager, T. Bridges, K. Gardner, S. Rogers, D. Belluck, and A. Meyer, "Multicriteria decision analysis: A comprehensive decision approach for management of contaminated sediments," Risk Analysis, vol. 26, 2006, pp. 61-78.
66. J. Wang, Y. Jing, C. Zhang, and J. Zhao, "Review on multi-criteria decision analysis aid in sustainable energy decision-making," Renewable and Sustainable Energy Reviews, vol. 13, 2009, pp. 2263-2278.
67. K. Steele, Y. Carmel, J. Cross, and C. Wilcox, "Uses and misuses of multicriteria decision analysis (MCDA) in environmental decision making," Risk Analysis, vol. 29, 2009, pp. 26-33.
68. I. Yuksel and M. Dagdeviren, "Using the analytic network process (ANP) in a SWOT analysis - A case study for a textile firm," Information Sciences, vol. 177, 2007, pp. 3364-3382.
69. Umm-E-habiba and S. Asghar, "A survey on multi-criteria decision making approaches," 2009 International Conference on Emerging Technologies, ICET 2009, October 19, 2009 - October 20, 2009, Islamabad, Pakistan: IEEE Computer Society, 2009, pp. 321-325.
70. N. Caterino, I. Iervolino, G. Manfredi, and E. Cosenza, "Comparative analysis of multi-criteria decision-making methods for seismic structural retrofitting," Computer-Aided Civil and Infrastructure Engineering, vol. 24, 2009, pp. 432-445.
71. S. Opricovic and G. Tzeng, "Extended VIKOR method in comparison with outranking methods," European Journal of Operational Research, vol. 178, 2007, pp. 514-529.
72. A. Milani and A. Shanian, "Gear material selection with uncertain and incomplete data. Material performance indices and decision aid model," International Journal of Mechanics and Materials in Design, vol. 3, 2006, pp. 209-222.
73. C. Zopounidis and M. Doumpos, "Multicriteria classification and sorting methods: A literature review," European Journal of Operational Research, vol. 138, 2002, pp. 229-246.
74. A. Kangas, J. Kangas, and J. PykÃ¤lÃ¤inen, "Outranking methods as tools in strategic natural resources planning," Silva Fennica, vol. 35, 2001, pp. 215-227.
75. P. Chatterjee, V. Athawale, and S. Chakraborty, "Selection of materials using compromise ranking and outranking methods," Materials and Design, vol. 30, 2009, pp. 4043-4053.
76. S. Cheng, C.W. Chan, and G.H. Huang, "Using multiple criteria decision analysis for supporting decisions of solid waste management," Journal of Environmental Science and Health - Part A Toxic/Hazardous Substances and Environmental Engineering, vol. 37, 2002, pp. 975-990.
77. S. Gilliams, D. Raymaekers, B. Muys, and J. Van Orshoven, "Comparing multiple criteria decision methods to extend a geographical information system on afforestation," Computers and Electronics in Agriculture, vol. 49, 2005, pp. 142-158.
78. M. Chu, J. Shyu, G. Tzeng, and R. Khosla, "Comparison among three analytical methods for knowledge communities group-decision analysis," Expert Systems with Applications, vol. 33, 2007, pp. 1011-1024.
79. S. Opricovic and G. Tzeng, "Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS," European Journal of Operational Research, vol. 156, 2004, pp. 445-455.
80. M. Behzadian, R. Kazemzadeh, A. Albadvi, and M. Aghdasi, "PROMETHEE: A comprehensive literature review on methodologies and applications," European Journal of Operational Research, vol. 200, 2010, pp. 198-215.

81. W. Tsai, J. Leu, J. Liu, S. Lin, and M.J. Shaw, “A MCDM approach for sourcing strategy mix decision in IT projects,” Expert Systems with Applications, vol. 37, 2010, pp. 3870-3886.
82. I. Gorton and A. Liu, “Software component quality assessment in practice: successes and practical impediments,” Proceedings of the 24th International Conference on Software Engineering. ICSE 2002, 19-25 May 2002, New York, NY, USA: ACM, 2002, pp. 555-8.
83. J. Winter, K. Ronkko, M. Ahlberg, M. Hinely, and M. Hellman, “Developing Quality through Measuring Usability--The UTUM Test Package,” Software Quality, 2007. WoSQ'07: ICSE Workshops 2007. Fifth International Workshop on, 2007, p. 2.
84. H. Washizaki, Y. Kobayashi, H. Watanabe, E. Nakajima, Y. Hagiwara, K. Hiranabe, and K. Fukuda, “Experiments on quality evaluation of embedded software in Japan robot software design contest,” Proceedings of the 28th international conference on Software engineering, Shanghai, China: ACM, 2006, pp. 551-560.
85. T. Bakota, Á. Beszédes, R. Ferenć, and T. Gyimóthy, “Continuous software quality supervision using SourceInventory and Columbus,” Companion of the 30th international conference on Software engineering, Leipzig, Germany: ACM, 2008, pp. 931-932.
86. L. Qu and Y. Chen, “QoS ontology based efficient web services selection,” 2009 16th International Conference on Management Science and Engineering, ICMSE 2009, September 14, 2009 - September 16, 2009, Moscow, Russia: IEEE Computer Society, 2009, pp. 45-50.
87. H. Washizaki, R. Namiki, T. Fukuoka, Y. Harada, and H. Watanabe, “A framework for measuring and evaluating program source code quality,” Product-Focused Software Process Improvements. 8th International Conference, PROFES 2007, 2-4 July 2007, Berlin, Germany: Springer-Verlag, 2007, pp. 284-99.
88. G. Buyukozkan, D. Ruan, and O. Feyzioglu, “Evaluating e-Learning Web site quality in a fuzzy environment,” International Journal of Intelligent Systems, vol. 22, 2007, pp. 567-586.
89. S. Barney, C. Wohlin, and A. Aurum, “Balancing software product investments,” 2009 3rd International Symposium on Empirical Software Engineering and Measurement, ESEM 2009, October 15, 2009 - October 16, 2009, Lake Buena Vista, FL, United states: IEEE Computer Society, 2009, pp. 257-268.
90. B. Wong and R. Jeffery, “Cognitive structures of software evaluation: a means-end chain analysis of quality,” Product Focused Software Process Improvement. Third International Conference, PROFES 2001. Proceedings, 10-13 Sept. 2001, Berlin, Germany: Springer-Verlag, 2001, pp. 6-26.
91. S.W. Cunningham and T.E. van der Lei, “Decision-making for new technology: A multi-actor, multi-objective method,” Technological Forecasting and Social Change, vol. 76, 2009, pp. 26-38.
92. B. Wong and R. Jeffery, “A framework for software quality evaluation,” Product Focused Software Process Improvement. 4th International Conference, PROFES 2002. Proceedings, 9-11 Dec. 2002, Berlin, Germany: Springer-Verlag, 2002, pp. 103-18.
93. S. Biffl, R. Mordinyi, and A. Schatten, “A model-driven architecture approach using explicit stakeholder quality requirement models for building dependable information systems,” 5th International Workshop on Software Quality, WoSQ 2007, May 20, 2007 - May 26, 2007, Minneapolis, MN, United states: Inst. of Elec. and Elec. Eng. Computer Society, 2007, p. IEEE Computer Society Technical Council on Software Engineering; The ACM Special Interest Group on Software Engineering (SIGSOFT).

94. T. Punter, A. Trendowicz, and P. Kaiser, “Evaluating evolutionary software systems,” Product Focused Software Process Improvement. 4th International Conference, PROFES 2002. Proceedings, 9-11 Dec. 2002, Berlin, Germany: Springer-Verlag, 2002, pp. 258-72.
95. M. Bombardieri and F. Fontana, “Software aging assessment through a specialization of the SQuaRE quality model,” 2009 ICSE Workshop on Software Quality (WoSQ 2009), 16 May 2009, Piscataway, NJ, USA: IEEE, 2009, pp. 33-8.
96. B. Yannou, “Towards a web-based collaborative weighting method in project,” Yasmine Hammamet, Tunisia: Institute of Electrical and Electronics Engineers Inc., 2002, pp. 488-493.
97. H. Washizaki, H. Hiraguchi, and Y. Fukazawa, “A metrics suite for measuring quality characteristics of Javabeans components,” Product-Focused Software Process Improvement. 9th International Conference, PROFES 2008, 23-25 June 2008, Berlin, Germany: Springer-Verlag, 2008, pp. 45-60.
98. H. Benestad, B. Anda, and E. Arisholm, “Assessing software product maintainability based on class-level structural measures,” Product-Focused Software Process Improvement. 7th International Conference, PROFES 2006. Proceedings, 12-14 June 2006, Berlin, Germany: Springer-Verlag, 2006, pp. 94-111.
99. O. Mizuno, S. Ikami, S. Nakaichi, and T. Kikuno, “Fault-prone filtering: Detection of fault-prone modules using spam filtering technique,” 1st International Symposium on Empirical Software Engineering and Measurement, ESEM 2007, September 20, 2007 - September 21, 2007, Madrid, Spain: Inst. of Elec. and Elec. Eng. Computer Society, 2007, pp. 374-383.
100. M. Klas, F. Elberzhager, and H. Nakao, “Managing software quality through a hybrid defect content and effectiveness model,” 2nd International Symposium on Empirical Software Engineering and Measurement, ESEM 2008, October 9, 2008 - October 10, 2008, Kaiserslautern, Germany: Association for Computing Machinery, 2008, pp. 321-323.
101. R. Kazman, L. Bass, G. Abowd, and M. Webb, “SAAM: a method for analyzing the properties of software architectures,” Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on, 1994, pp. 81-90.
102. R. Plosch, H. Gruber, A. Hentschel, C. Korner, G. Pomberger, S. Schiffer, M. Saft, and S. Storck, “The EMISQ method - Expert based evaluation of internal software quality,” 31st Annual IEEE Software Engineering Workshop, SEW-31 2007, March 6, 2007 - March 8, 2007, Columbia, MD, United states: Inst. of Elec. and Elec. Eng. Computer Society, 2007, pp. 99-108.
103. I. Vlahavas, I. Stamelos, I. Refanidis, and A. Tsoukias, “ESSE: an expert system for software evaluation,” Knowledge-Based Systems, vol. 12, 1999, pp. 183-97.
104. J. Deprez and S. Alexandre, “Comparing assessment methodologies for Free/Open Source Software: OpenBRR and QSOS,” Product-Focused Software Process Improvement. 9th International Conference, PROFES 2008, 23-25 June 2008, Berlin, Germany: Springer-Verlag, 2008, pp. 189-203.
105. S. Neti and H.A. Muller, “Quality criteria and an analysis framework for self-healing systems,” ICSE 2007 Workshops: International Workshop on Software Engineering for Adaptive and Self-Managing Systems, SEAMS'07, May 20, 2007 - May 26, 2007, Minneapolis, MN, United states: Inst. of Elec. and Elec. Eng. Computer Society, 2007, p. IEEE Computer Society; SIGSOFT.
106. Shang-Wen Cheng, D. Garlan, and B. Schmerl, “Evaluating the effectiveness of the rainbow self-adaptive system,” 2009 ICSE Workshop on Software Engineering for Adaptive and

Self-Managing Systems (SEAMS 2009), 16-24 May 2009, Piscataway, NJ, USA: IEEE, 2009, pp. 132-41.

107. E. Raffoul, K. Domínguez, M. Pérez, L.E. Mendoza, and A.C. Grimán, “Quality model for the selection of floss-based issue tracking system,” Proceedings of the IASTED International Conference on Software Engineering, Innsbruck, Austria: ACTA Press, 2008, pp. 43-49.

108. S. Morasca, “Subjective assessment of the mutual influence of ISO 9126 software qualities: an empirical study,” SEKE 2008. The 20th International Conference on Software Engineering & Knowledge Engineering, 1-3 July 2008, Skokie, IL, USA: Knowledge Systems Institute Graduate School, 2008, pp. 297-302.

109. N. Thomaidis, N. Nikitakos, and G. Dounias, “The evaluation of information technology projects: A fuzzy multicriteria decision-making approach,” International Journal of Information Technology and Decision Making, vol. 5, 2006, pp. 89-122.

110. J. Brunet, D. Guerrero, and J. Figueiredo, “Design tests: An approach to programmatically check your code against design rules,” 2009 31st International Conference on Software Engineering, ICSE 2009, May 16, 2009 - May 24, 2009, Vancouver, BC, Canada: IEEE Computer Society, 2009, pp. 255-258.

111. E. Kornyshova and C. Salinesi, “MCDM Techniques Selection Approaches: State of the Art,” 2007.

11 APPENDIX

11.1 Appendix A: Terminology

Table 47: Terminologies used in this thesis report

Terms	Definition
Systematic Literature Review	"A means of identifying, evaluating and interpreting all available research relevant to a particular research question, topic area or phenomenon of interest" [6].
Multi-criteria decision analysis (MCDA)	MCDA is "an umbrella term to describe a collection of formal approaches which seek to take explicit account of multiple criteria in helping individuals or groups explore the decisions that matter" [46].
MCDA base methods	MCDA base methods are the original MCDA methods without any extension and integration with other approaches. For example, Analytic Hierarchy Process (AHP) is considered as elementary MCDA method. Besides, all extended version of base MCDA methods or hybrid MCDA are not classified as the elementary MCDA methods.
Hybrid MCDA methods	Hybrid MCDA methods are the methods where any two or more MCDA methods are integrated or the integrated methods derived from MCDA methods and other approaches (e.g.: Fuzzy Set theory) [14]. For example, Fuzzy AHP is a hybrid MCDA method that combines AHP with fuzzy set theory.
Decision problem	A decision problem designates a situation where a decision maker or a group of them have a list of known alternatives on hand and need to determine which one of the identified alternatives can satisfy their need or achieved their ultimate goal in the most optimum way.
Decision maker	Decision maker is defined as the individual or a group who is the owner of a specific decision problem [14, 40].
Alternatives [14]	This term is defined as the possible solution or options to a MCDA decision problem that the decision maker(s) has to decide within [14, 23, 44].
Criteria [14, 23]	In MCDA, the criteria are referred as the attributes or factors that describe a decision problem.
Threshold [14]	A value that is applied to account for the imprecision on the result of certain comparison. It allows building the equality between two

	alternatives assessed different in a given scale.
Consequence	Consequence is a MCDA terminology that indicates the actual performance of the alternative with respect to a decision criterion [44].
Score	A value to indicate the strength of preference in accordance to the consequence of criteria for an alternative [44].
Weight	A value to indicate the relative importance of a decision criterion in certain decision process or context [14, 45]. However, the concrete interpretation of this concept can be varied according to different MCDA method.
Ideal point	A point in the criterion space that possess the maximum value for each dimension [14, 44].
Preference relation	Preference relation is a binary relation that expresses the way an action is preferred against another one [14, 40]. Different scales can be used to express these preferences.
Deficit	The insufficiency or incapability of MCDA methods to fulfill an expected behavior.
Basic element	The conceptual component or aspects of MCDA methods that being frequently discussed in the publications.

11.2 Appendix B: Publications Selected for systematic review in MCDA

Table 48: List of selected publications in first review

No.	Publication Title	Author(s)
1	A comparison of two multi-criteria decision-making techniques	Farnaz Akhavi, Caroline Hayes
2	A critical review of multi-criteria decision making methods with special reference to forest management and planning	Jayanath Ananda, Gamini Herath
3	A hybrid MCDM model for strategic vendor selection	Huan-Jyh Shyur, Hsu-Shih Shih
4	A MCDM approach for sourcing strategy mix decision in IT projects	Wen-Hsien Tsai, Jun-Der Leu, Jau-Yang Liu, Sin-Jin Lin, Michael J. Shaw
5	A survey on multi-criteria decision making approaches	Umm-e-Habiba, Sohail Asghar
6	Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study	Anil Jadhav, Rajendra Sonar
7	Application of multicriteria decision analysis in environmental decision making.	Gregory A. Kiker, Todd S. Bridges, Arun Varghese, Thomas P. Seager, Igor Linkov
8	Application of multi-criteria decision making to sustainable energy planning – A review	S.D. Pohekar, M. Ramachandran
9	Comparative analysis of multi-criteria decision-making methods for seismic structural retrofitting	N. Caterino, I. Iervolino, G. Manfrendi, E. Cosenza
10	Comparing MCDA aggregation methods in constructing composite indicators using the Shannon-Spearman measure	P. Zhou, B. W. Ang
11	Comparing multiple criteria decision methods to extend a geographical information system on afforestation	S. Gilliams, D. Raymaekers, B. Muys, J. Van Orshoven
12	Comparison among three analytical methods for knowledge communities group-decision analysis	Mei-Tai Chu, Joseph Shyu, Gwo-Hshiung Tzeng, Rajiv Khosla

13	Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS	Serafim Opricovic, Gwo-Hshiung Tzeng
14	Contractor selection using the analytic network process	Eddie W.L. Cheng, Heng Li
15	Criteria for evaluating group decision-making methods	Kirti Peniwati
16	Extended VIKOR method in comparison with outranking methods	Serafim Opricovic, Gwo-Hshiung Tzeng
17	From comparative risk assessment to multi-criteria decision analysis and adaptive management: Recent developments and applications	I Linkov, F.K. Satterstrom, G. Kiker, C. Batchelor, T. Bridges, E. Ferguson
18	Fuzzy AHP to determine the relative weights of evaluation criteria and Fuzzy TOPSIS to rank the alternatives	Fatemeh Torfi, Reza Zanjirani Farahani, Shabnam Rezapour
19	Gear material selection with uncertain and incomplete data. Material performance indices and decision aid model	A S Milani, A. Shanian
20	GIS-based multicriteria decision analysis: a survey of the literature	Jacek Malczewski
21	Incorporating multiple criteria into the design of conservation area networks: A minireview with recommendations	Alexander Moffett, Sahotra Sarkar
22	Making forestry decisions with multiple criteria: A review and an assessment	Luis Diaz-Balteiro, Carlos Romero
23	MCDA and preference disaggregation in Group Decision Support Systems	Nikolaos F. Matsatsinis, Andreas P. Samaras
24	MCDM Techniques Selection Approaches: State of the Art	E. Kornyshova, C. Salinesi
25	Multicriteria classification and sorting methods: A literature review	Constantin Zopounidis, Michael Doumpos
26	Multi-criteria decision analysis and environmental risk assessment for nanomaterials	Igor Linkov, F. Kyle Satterstrom, Jeffery Steevens, Elizabeth Ferguson, Richard C. Pleus
27	Multi-criteria decision analysis in natural resource management: A critical review of methods and new modeling paradigms	G A Mendoza, H. Martins
28	Multicriteria decision analysis: A comprehensive decision approach for management of contaminated	I Linkov, F.K. Satterstrom, G. Kiker, T.P. Seager, T. Bridges,

	sediments	K.H. Gardner, S.H. Rogers, D.A. Belluck, A. Meyer
29	Multi-Criteria Decision-Making: The Intersection of Search, Preference Tradeoff, and Interaction Visualization Processes	Piero P. Bonissone
30	New modeling paradigms in using multi-criteria decision analysis for sustainable forest management	G A. Mendoza, H. Martins
31	Outranking methods as tools in strategic natural resources planning	Annika Kangas, Jyrki Kangas, Jouni Pykäläinen
32	Preference disaggregation: 20 Years of MCDA experience	Eric Jacquet-Lagrèze, Yannis Siskos
33	PROMETHEE: A comprehensive literature review on methodologies and applications	Majid Behzadian, R.B. Kazemzadeh, A. Albadvi, M. Aghdasi
34	Review on multi-criteria decision analysis aid in sustainable energy decision-making	Jiang-Jiang Wang, You-Yin Jing, Chun-Fa Zhang, Jun-Hong Zhao
35	Selection of materials using compromise ranking and outranking methods	Prasenjit Chatterjee, Vijay Manikrao Athawale, Shankar Chakraborty
36	Use of multicriteria decision analysis methods for energy planning problems	Espen Løken
37	Uses and misuses of multicriteria decision analysis (MCDA) in environmental decision making	Katie Steele, Yohay Carmel, Jean Cross, Chris Wilcox
38	Using multiple criteria decision analysis for supporting decisions of solid waste management	Steven Cheng, Christine W. Chan, Guo H. Huang
39	Using the analytic network process (ANP) in a SWOT analysis – A case study for a textile firm	Ihsan Yüksel, Metin Dağdeviren

11.3 Appendix C: Publications Selected for systematic review in SQA

Table 49: List of selected publications in second review

No.	Publication Title	Author(s)
1	A framework for measuring and evaluating program source code quality	Hironori Washizaki, Rieko Namiki, Tomoyuki Fukuoka, Yoko Harada, Hiroyuki Watanabe
2	A framework for software quality evaluation	Bernard Wong, Ross Jeffery
3	A metrics suite for measuring quality characteristics of Javabeans components	Hironori Washizaki, Hiroki Hiraguchi, Yoshiaki Fukazawa
4	A model-driven architecture approach using explicit stakeholder quality requirement models for building dependable information systems	Steffan Biffl, Richard Mordinyi, Alexander Schatten
5	Assessing software product maintainability based on class-level structural measures	Hans Christian Benestad, Bente Anda, Erik Arisholm
6	Balancing software product investments	Sebastian Barney, Claes Wohlin
7	Cognitive structures of software evaluation: a means-end chain analysis of quality	Bernard Wong, Ross Jeffery
8	Comparing assessment methodologies for Free/Open Source Software: OpenBRR and QSOS	Jean-Christophe Deprez, Simon Alexandre
9	Design tests: An approach to programmatically check your code against design rules	João Brunet, Dalton Guerrero, Jorge Figueiredo
10	Evaluating evolutionary software systems	Teade Punter, Adam Trendowicz, Peter Kaiser
11	Evaluating the effectiveness of the rainbow self-adaptive system	Shan-Wen Cheng, David Garlan, Bradley Schmerl
12	Fault-prone filtering: Detection of fault-prone modules using spam filtering technique	Osamu Mizuno, Shiro Ikami, Shuya Nakaichi, Tohru Kikuno
13	Managing software quality through a hybrid defect content and effectiveness model	Michael Kläs, Frank Elberzhager, Haruka Nakao
14	Software aging assessment through a specialization of the SQuaRE quality model	Michele Bombardieri, Francesca Arcelli Fontana
15	Software component quality assessment in practice: successes and practical impediments	Ian Gorton, Anna Liu

16	Subjective assessment of the mutual influence of ISO 9126 software qualities: an empirical study	S. Morasca
17	Developing Quality through Measuring Usability--The UTUM Test Package	Jeff Winter, Kari Rönkkö, Mårten Ahlberg, Mark Hinelz, Mats Hellman
18	Quality Criteria and an Analysis Framework for Self-Healing Systems	Sangeeta Neti, Hausi A. Müller
19	SAAM: a method for analyzing the properties of software architectures	R. Kazman, L. Bass, G. Abowd, M. Webb
20	Experiments on quality evaluation of embedded software in Japan robot software design contest	Hironori Washizaki, Yasuhide Kobayashi, Hiroyuki Watanabe
21	Continuous software quality supervision using SourceInventory and Columbus	Tibor Bakota, Árpád Beszédes, Rudolf Ferenc, Tibor Gyimóthy
22	Quality model for the selection of floss-based issue tracking system	Eduardo Raffoul, Kenyer Domínguez, María Pérez, Luis E. Mendoza, Anna C. Grimán
23	The EMISQ method - Expert based evaluation of internal software quality	R. Plösch, H. Gruber, A. Hentschel, Ch. Körner, G. Pomberger, S. Schiffer, M. Saft, S. Storck
24	An intelligent approach to ERP software selection through fuzzy ANP	Z. Ayağ, R.G. Őzdemir
25	COTS evaluation using modified TOPSIS and ANP	Huan-Jyh Shyur
26	Making decisions: Using Bayesian nets and MCDA	N. Fenton, M. Neil
27	Evaluating e-Learning Web site quality in a fuzzy environment	Gülçin Büyüközkan, Da Ruan, Orhan Fezyioğlu
28	Evaluation of software development projects using a fuzzy multi-criteria decision approach	Gülçin Büyüközkan, Da Ruan
29	QoS ontology based efficient web services selection	Qu Li-li, Chen Yan
30	The evaluation of information technology projects: A fuzzy multicriteria decision-making approach	Nikolaos S. Thomaidis, Nikitas Nikitakos, Georgios D. Dounias
31	The Evaluation of Software Trustworthiness with FAHP and FTOPSIS Methods	Li Shi, Shanlin Yang
32	Towards a web-based collaborative weighting method in project	Bernard Yannou

33	Decision-making for new technology: A multi-actor, multi-objective method	Scott W. Cunningham, Telli E.van der Lei
34	ESSE: an expert system for software evaluation	I Vlahavas, I. Stamelos, I. Refanidis, A. Tsoukiàs
35	A simplified measurement scheme for software quality	Chang Che-Wei, Wu Cheng-Ru, Lin Hung-Lung

11.4 Appendix D: Overview of the Example Scenario

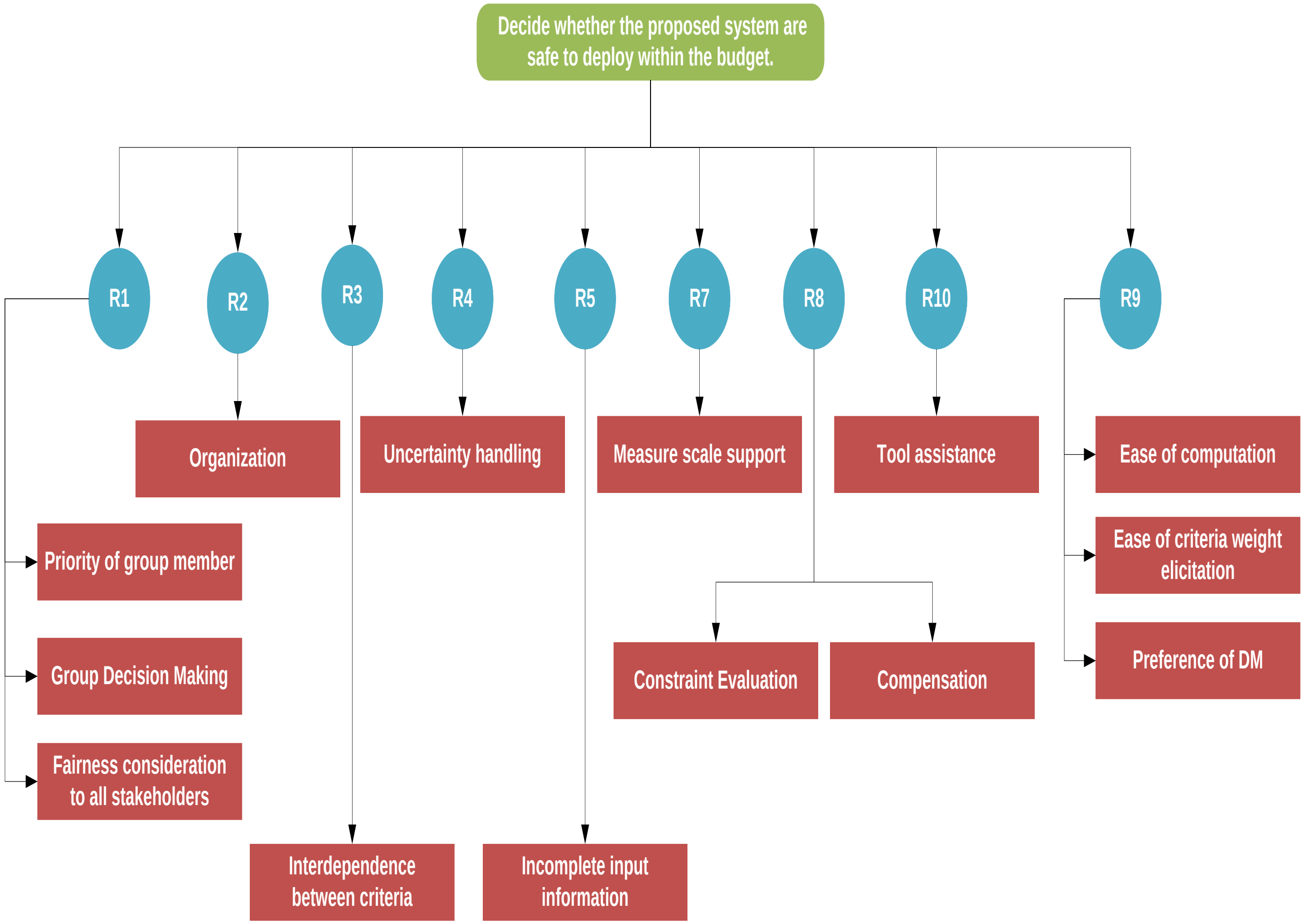

Figure 19: Overview of the example scenario

11.5 Appendix E: Distribution of Priority Points for MCDA Methods

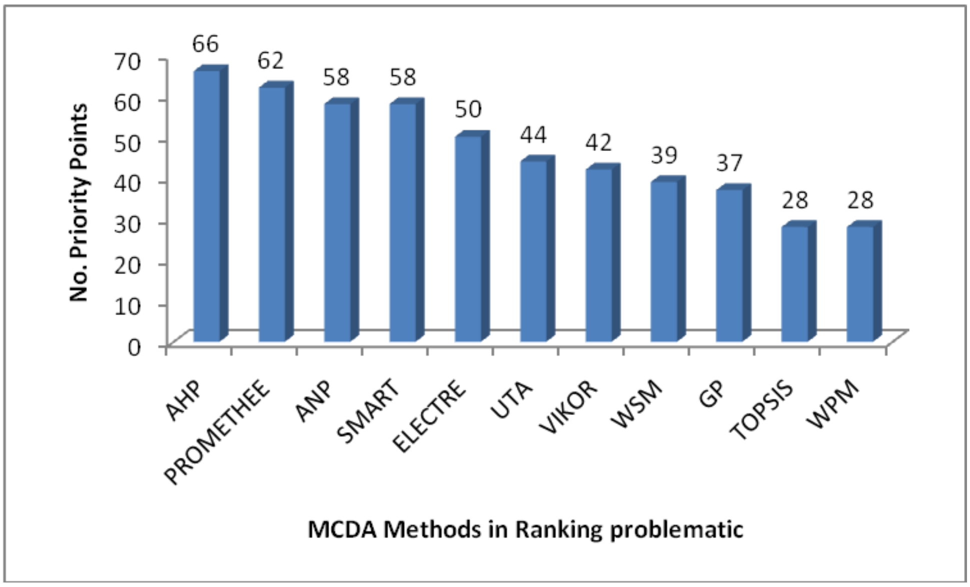

Figure 20: Distribution of priority points for MCDA methods supporting ranking problematic

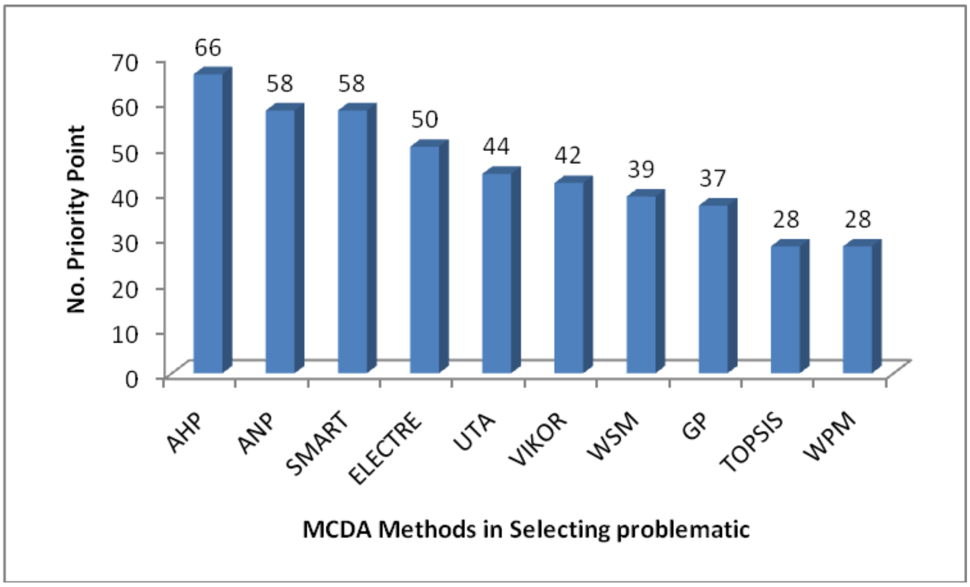

Figure 21: Distribution of priority points for MCDA methods supporting selecting problematic

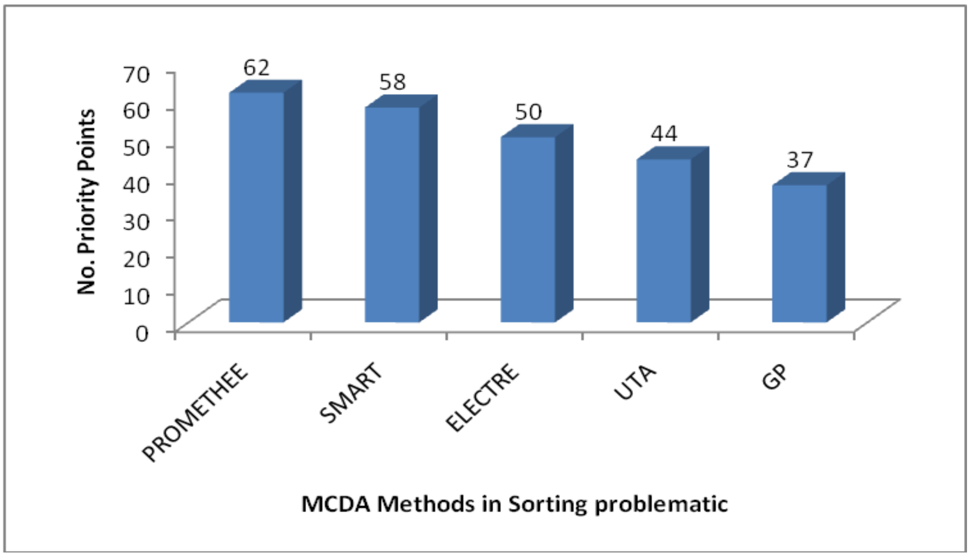

Figure 22: Distribution of priority points for MCDA methods supporting sorting problematic

11.6 Appendix F: Basic Elements of MCDA Methods

Table 50: List of basic elements of MCDA methods

No.	ID	Basic elements
1	U1	Ease of use
2	U2	Uncertainty handling
3	U3	Organization
4	U4	Group Decision Making
5	U5	Effort spend in decision analysis
6	U6	Judgment conflict resolution
7	U7	Tool assistance
8	U8	Applicability to group decision making [43]
9	U9	Dimension of decision problem
10	U10	Leadership assurance [43]
11	U11	Learning
12	U12	Scope [43]
13	U13	Assistance of alternative development [43]
14	U14	Priority of group members inclusion
15	U15	Fairness consideration to all stakeholders [43]
16	U16	Implementation cost
17	U17	Preference of decision maker
18	U18	Incomplete input information
19	C1	Nature of criteria
20	C2	Optimum size of criteria.
21	C3	Supporting criteria weight

22	C4	Non-commensurable criteria handling
23	C5	Interdependence between criteria
24	C6	Nature of criteria weight in the MCDA method
25	C7	Supported measurement scale
26	C8	Dimension of criteria
27	C9	Ease of criteria weight elicitation
28	C10	Effort in criteria change
29	A1	Optimum size of alternative
30	A2	Nature of alternative
31	A3	Interdependence between alternatives
32	A4	Alternative change
33	P1	Aggregation
34	P2	Types of problematic
35	P3	Preference model
36	P4	Alternative evaluation
37	P5	Completeness
38	P6	Required model parameter setting
39	P7	Compensation [40]
40	P8	Rank reversal
41	P9	Constraint inclusion
42	P10	Risk evaluation
43	P11	Support relation between decision level
44	P12	Aggregation condition
45	P13	Moment to include preference
46	P14	Validity of evaluation
47	P15	Ambiguity of evaluation result
48	P16	Transitivity
49	P17	"Psychophysical applicability" [43]

11.7 Appendix G: Selection Requirements

Table 51: Selection Requirements of MCDA-SQA selection framework

ID	Selection requirements
R1	The method shall be comprehensable.
R2	The method shall be able to aggregate different quality attributes.
R3	The method shall support group decision making.
R4	The method shall be flexible in modeling the quality.

R5	The method shall include the consideration about the values and consequences of applying the software product.
R6	The method shall be able to perform evaluation even the information required is not complete.
R7	The method shall provide an unambigious computation procedure which all decision makers should understand it in the same way.
R8	The method shall have repository to store evaluation scores for future reuse purpose.
R9	The method shall require less assessment effort.
R10	The method shall provide a common benchmark system.
R11	The method shall facilitate quality trade-offs.
R12	The method shall be reusable or repeatable.
R13	The method shall be able to handle uncertain data.
R14	The method shall be able to handle interdependencies between the quality attributes
R15	The method shall have been implemented in a software tool to assist or automate its manual computation.
R16	The method shall express the reliability of the stakeholders' opinions.
R17	The method shall include and express different judgements from the stakeholders.
R18	The method should be applicable for the standard quality model.
R19	The method shall provide reliable assessment result.
R20	The method shall facilitate the learning mechanism
R21	The method shall support software improvement.

11.8 Appendix H: Mapping of Selection Requirement to Related MCDA Method Aspects

Table 52: Relationship between selection requirement and basic elements of MCDA method

Requirement ID	Selection criteria	Characteristics
R1. Comprehensable.	U1 : Ease of use	Low, medium, high, very high
	U17: Preference of decision maker	Low, medium, high
	C10: Ease of criteria weight elicitation	Low, medium, high
R2. Aggregate different quality attributes.	P1 : Aggregation	Single Synthesizing criterion, outranking
R3. Group decision making.	U4 : Group Decision Making	Support (yes), not support (no)
	U6 : Judgement Conflict	Support (yes), not support (no)

R4. Flexible in modeling the quality.	-	-
R5. Values and consequences consideration.	C9 : Dimension of criteria	Low, medium, high
	P9 : Constraint evaluation	Support (yes), not support (no)
R6. Evaluation under incomplete input information.	U18: Incomplete input information	Support (yes), not support (no)
R7. Unambiguious result.	P15: Ambiguity	Support (yes), not support (no)
R8. Repository.	-	-
R9. Assessment effort.	U5 : Effort spend in decision analysis	Low, medium, high
R10. Common benchmarking.	Supported by all MCDA methods.	-
R11. Quality trade-offs.	P7 : Compensation	Total compensatory, partial compensatory, non-compensatory
R12. Repeatable.	C10: Effort when criteria change	Low, medium, high
	A4 : Effort spend when alternative change	Low, medium, high
R13. Uncertain data.	U2 : Uncertain handling	Support (yes), not support (no)
R14. Interdependencies between the quality attributes	C5 : Interdependence between criteria	Support (yes), not support (no)
R15. Software tool	U7 : Tool assistance	Support (yes), not support (no)
	U16 : Implementation cost	Low, medium, high
R16. Reliability of stakeholders.	U14 : Priority of group member inclusion	None, low, medium, high, very high
R17. Stakeholders' judgements inclusion.	U15 : Fairness consideration to all stakeholder	None, low, medium, high
R18. Quality modeling.	U3 : Organization	None, tree structure, network structure
R19. Reliable assessment result.	P14: Validity	Low, medium, high, very high
R20. Learning mechanism.	U11: Learning	Low, medium, high
R21. Softwar improvement.	-	-