Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Tormod Mork Müller

# Enhancing Vendor Selection in Software Ecosystems: A Decision-Making Tool

Master's thesis in Applied Computer Science
Supervisor: Anshul Rani
Co-supervisor: Deepti Mishra
June 2024

NTNU
Norwegian University of
Science and Technology

Tormod Mork Müller

# Enhancing Vendor Selection in Software Ecosystems: A Decision-Making Tool

Master's thesis in Applied Computer Science
Supervisor: Anshul Rani
Co-supervisor: Deepti Mishra
June 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

# Abstract

This thesis proposes a decision-support tool for software vendor selection in software ecosystems, expanding a foundational framework to handle decision-maker uncertainty and providing an extensive review of non-functional requirements as selection criteria.

The software vendor selection process is often complex, ad-hoc, and prone to errors. Although numerous methods have been proposed to assist decision-makers, most lack adaptability and integration into existing processes. To address this, user-friendly interfaces and tools are suggested as potential solutions.

The tool developed in this thesis work builds on a state-of-the-art vendor analysis and selection framework, which is scaled to utilize Fuzzy Set Theory to manage uncertainty during decision-making. This enables decision-makers to more effectively handle imprecise data, subjective judgments, and multiple conflicting objectives, which often are present in these processes. Furthermore, the tool incorporates user-friendly interfaces and functionalities, drawing on feedback from practitioners and literature to ensure its adaptability and integration into existing processes.

Additionally, a baseline set of eight non-functional requirements are addressed as selection criteria, along with tailored definitions for the context. These criteria and definitions pave new paths for how non-functional requirements should be considered as selection criteria in software vendor selection for software ecosystems.

# Sammendrag

Denne oppgaven presenterer et beslutningsstøtteverktøy for valg av programvareleverandører i programvaremiljøer. Dette gjøres ved å utvide et grunnleggende rammeverk for å håndtere beslutningstakeres usikkerhet og gi en omfattende gjennomgang av ikke-funksjonelle krav som valgskriterier.

Valg av programvareleverandører er ofte en kompleks, ad hoc-prosess som er utsatt for feil. Selv om det tidligere har blitt foreslått mange metoder for å bistå beslutningstakere, mangler de fleste tilpasningsevne og integrasjon i eksisterende prosesser. For å løse dette foreslås brukervennlige brukergrensesnitt og verktøy som mulige løsninger.

Verktøyet utviklet i denne oppgaven bygger på et avansert rammeverk for leverandøranalyse og -valg, som er utvidet til å benytte Fuzzy Set Theory for å håndtere usikkerhet under beslutningstaking. Dette gjør det mulig for beslutningstakere å håndtere upresise data, subjektive vurderinger og flere motstridende mål mer effektivt, som ofte er til stede i disse prosessene. Videre inkluderer verktøyet brukervennlige grensesnitt og funksjonaliteter, basert på tilbakemeldinger fra praktiserende beslutningstakere og litteratur for å sikre dets tilpasningsevne og integrasjon i eksisterende prosesser.

I tillegg adresseres et sett på åtte ikke-funksjonelle krav som valgskriterier, sammen med skreddersydde definisjoner for den gitte konteksten. Disse kriteriene og definisjonene åpner nye veier for hvordan ikke-funksjonelle krav bør vurderes som valgskriterier i valg av programvareleverandører for programvaremiljøer.

# Acknowledgements

I would like to express my sincere gratitude to my supervisors, Anshul Rani and Deepti Mishra, for introducing me to the field of software vendor selection within software ecosystems and their guidance. Their support throughout this journey has been instrumental in helping me reach this milestone. A special thanks to Anshul for her continuous support, patience and learning me all she knows about the topic.

I would also like to extend my appreciation to the practitioners who generously cleared their busy schedules to evaluate the decision-support tool and share their valuable experiences.

Lastly, I would like to thank all my close supporters for making this journey as enjoyable and fulfilling as it has been.

*Tormod Mork Müller*

# Contents

# Figures

# Tables

# Chapter 1

# Introduction

The chapter firstly introduces the software ecosystems along with a motivating example of why research within the domain is deemed crucial. Afterwards, research questions and objectives are set to conduct the thesis work. Lastly, the planned contributions and the remaining thesis outline is presented.

## 1.1 Problem Description

The concept of Software Ecosystems (SECO) involves decentralized development where companies may outsource their product development either entirely or partially to one or more vendors [1]. The software ecosystem not only includes the participating companies but also includes the intricate interactions among different entities [2, 3] such as businesses, software service providers, and customers.

The entities involved in such ecosystems are often connected to a shared platform, collaborating to address challenges and pursue common objectives [4–6]. The structure of collaboration within these entities varies widely, reflecting different strategic approaches. For instance, software ecosystems often differ in how companies choose to collaborate with vendors. In a single-vendor ecosystem, as illustrated in Figure 1.1, the keystone company of the ecosystem, collaborates exclusively with one vendor. While this approach offers benefits like streamlined management, it also carries risks due to over-reliance on a single entity [7], which in turn can lead to issues such as limited innovation and vendor lock-in. In contrast, multi-vendor ecosystems or multi-sourcing [7] involve collaboration among

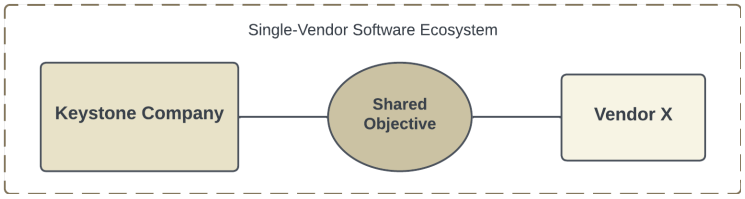


**Figure 1.1:** Single-Vendor Ecosystem

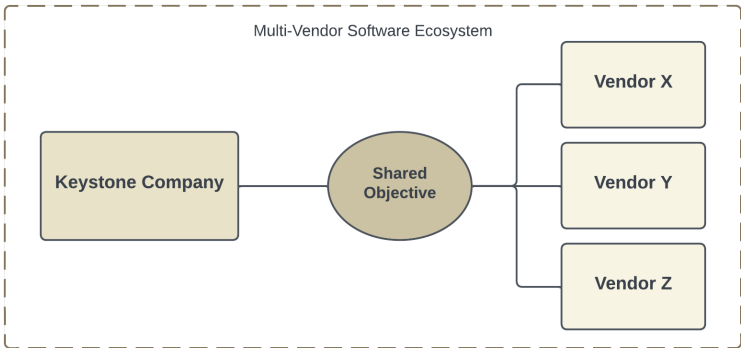


**Figure 1.2:** Multi-Vendor Ecosystem

multiple entities, which fosters innovation and diversify the risk [7] but can also lead to challenges in coordination and potential conflicts among the participants [5] (Illustrated in Figure 1.2). These dynamics raise a critical question: *Which vendor should the keystone companies or ecosystem as a whole choose to integrate into the ecosystem, and how can the best decision be reached?*

In a multi-vendor ecosystem, vendor selection transcends meeting the requirements of a single keystone company, extending to align with the collective needs and strategic objectives of the entire ecosystem [2]. This selection process is critical to the success of both the outsourcing company and vendors [1]. Incorrect vendor selection can significantly affect the quality of the product delivered and hence, reputation among stakeholders and customers, and lead to its failure [1]. The complexities of the decision-making process are compounded by the intricate interactions among different ecosystem entities [8].

The decision on vendor selection is typically made by the keystone company's decision-makers [1]. However, this decision-making process is characterized as complex, ad-hoc, and manual [5], adding layers of complexity to the system and creating barriers to ecosystem adoption [5] among companies. Despite the involvement of dedicated personnel, the decision-making process remains largely subjective and lacks transparency [1, 9], which can undermine trust and efficacy, and prove especially difficult when considering non-functional requirements (NFRs) as selection criteria [10]. Despite these requirements' crucial role, they are often overlooked or inadequately addressed. Adding to the complexities of these decision-making processes is the inherent uncertainty in evaluating NFRs and other selection criteria, which are crucial for ensuring that the software and vendor aligns with the broader operational and strategic goals of the ecosystem. This uncertainty arises from the subjective nature of these selection criteria and the often incomplete or imprecise data available during the decision-making process [9]. This in turn makes it difficult to assess vendors accurately and consistently.

Despite numerous efforts to streamline the decision-making process with various frameworks and methodologies [1, 11, 12], most approaches fall short in us-

ability and adaptability. To the best of the author's knowledge, existing literature is limited to theoretical vendor analysis and selection frameworks. These frameworks lack a user-friendly interface or a platform for the outsourcing companies to integrate into the existing organizational processes to support decision-makers. As a result, many decision-makers continue to rely on their current manual and ad-hoc methods [1, 5] in absence of structured and streamlined measures for the decision-making process.

## 1.2 Motivation

To motivate the importance of successful software vendor selection, a motivating case of what's referred to as the British Post Office scandal will be briefly introduced. While it can be motivated both from the company and the vendor's perspective, in this motivation it will be viewed from the outsourcing company's perspective.

In 1999, the Horizon system, developed by *Fujitsu Limited*[1], was selected by the UK Post Office[2], with pilots running from 1996 [13]. This system was chosen to replace outdated methods with what was referred to as a more accurate and reliable processing of transactions. Instead, the system ended up being full of technical issues that led to numerous accounting inaccuracies [14]. These inaccuracies resulted in severe financial discrepancies being recorded in the accounts of more than 900 sub-postmasters [14]. Regardless of Post Office being aware or unaware of the system's faults, they attributed a total of 700 of these discrepancies to theft, fraud, and misconduct, leading to the wrongful prosecution of several sub-postmasters and their employees [15]. Over time, these issues sparked a massive legal battle as affected sub-postmasters fought to clear their names [16]. The ensuing controversy exposed significant failures in oversight and accountability, raising questions about the due diligence (vendor analysis) conducted during the system's selection and the subsequent handling of its flaws. The consequences have proved enormous with this single faulty choice of software ultimately leading to ruining the victims lives [15], so much even some committing suicide [17], and enormous economical consequences. The final cost of compensations is expected to exceed £1 billion [14, 18], with some sources stating that the UK government has already set aside £1.27 billion of compensation to pay out to victims of the Post Office scandal [16], not to talk about the continuous growing legal cost and damaged reputation both for the Post Office, the UK government and Fujitsu who faced a $1 billion market value drop amid the Post Office scandal [7]. As Fujitsu currently has more than £2 billion worth of contracts with the UK government, holding the status of a key strategic supplier [17] it is unknown which consequences this will face if the UK government stands it place in their promise of pursuing the manufacturer of the Horizon IT system if an inquiry finds it is to

[1] `https://www.fujitsu.com/global/about/`

[2] `https://en.wikipedia.org/wiki/Post_Office_Limited`

blame [18].

The Post Office Horizon scandal highlights the severe consequences that can come from not carefully choosing software vendors. This example shows that a poor decision can lead to huge financial losses, wrongful accusations, and serious damage to reputations and in worst case lost lives. The negative effects impacted not just the individuals directly involved but also the broader reputations and financial health of the UK Post Office, the government, and Fujitsu. This situation makes it clear how important it is to thoroughly check and oversee software and vendors during these processes, ensuring all aspects are carefully considered. It demonstrates why there's a strong need to develop reliable tools for helping make better vendor selection decisions, to avoid these kinds of problems in the future and to protect against the significant impacts and costs they can cause.

## 1.3 Scope and Context

In this thesis, the problem is approached from the perspective of the keystone company (Figure 1.2). Details on the *context and scope*, along with the *beneficiary* are presented in Table 1.1.

**Table 1.1:** Context and Scope

| | |
|---|---|
| **Context and Scope** | Keystone companies outsource their software needs either partially or entirely to one or more vendors. Increasing the usability and adaptability of vendor analysis and selection processes is aimed for while excluding details on the specifics of vendor analysis from the project scope. |
| **Beneficiary** | Dedicated personnel which are involved in the decision-making process, usually referred to as decision-makers. A decision-maker is usually someone with trust and extensive knowledge about the keystone company or ecosystem and its architecture. |

## 1.4 Research Questions

It is important to establish the perspective from which the problem is approached. This thesis addresses the problem from the keystone company's perspective (Figure 1.2), aiming to fill multiple gaps in the existing literature and practices relevant to these outsourcing companies.

Considering the motivation (Section 1.2) and the gaps identified in Chapter 2, the following research questions are set:

**RQ1** How are different NFRs addressed as selection criteria for software vendor selection in software ecosystems?

(a) Which NFRs are utilized as selection criteria?

(b) How can NFRs be interpreted to better assist decision-makers in this particular context?

**RQ2** Which criteria weighting strategies are effectively utilized within this context and how can these be integrated into the ICR framework?

**RQ3** How can decision-making processes be streamlined to enhance usability?

## 1.5 Planned Contributions

This thesis aims to address the formulated research questions with significant contributions. First, identifying and interpreting NFRs as selection criteria in the context of software vendor selection in software ecosystems, including identification of the most used ones and their definitions.

Second, the aim is to scale the ICR framework presented by Rani et al. [1] to enable decision-makers to effectively handle imprecise data, subjective judgments, and multiple conflicting objectives. Adding such a layer to the framework empowers decision-makers to make more informed, robust, and defensible decisions in situations characterized by ambiguity and uncertainty.

Lastly, the thesis aims at increasing the ICR framework's adaptability and usability by implementing an end-to-end decision-support tool for guiding decision-makers through their decision-making process in a systematic, automated and streamlined manner. Such an implementation aims at accommodating current ad-hoc and manual processes. The insights gained from the preliminary works on the low-fidelity user-interface design and process flow proved valuable to the industry and resulted in a research paper that has been accepted for publication (Appendix A), details given below:

- *Müller Tormod, Anshul Rani, Deepti Mishra. "A Tool for Software Vendor Analysis and Selection." Human-Centered Software Engineering: 10th International Working Conference, HCSE 2024*[3]*, Springer International Publishing, 2024 (Accepted)*.

## 1.6 Previous Work and Target Audience

Before proceeding with the rest of this thesis, it is important to clarify the previous work conducted on the thesis topic in other courses, along with the intended audience and the level of understanding required.

Some of the work presented throughout this thesis was conducted during various courses throughout the previous semester. These works are detailed in Appendix B and will not be repeatedly mentioned to maintain the flow and readability of the thesis.

The thesis is designed primarily for individuals familiar with or interested in software vendor selection processes, whether in a general context or specifically

[3]https://hcse2024.wordpress.com

within software ecosystems. Decision-makers or stakeholders with expertise in this field will find the most value, as the intricacies of these processes are complex and go beyond the scope of this thesis. However, anyone with a background in computer science should be able to grasp and appreciate the research presented.

## 1.7 Thesis Outline

The rest of the thesis is structured as follows:

**2 Background** chapter details concepts and terminologies which are required for a seamless reading experience. It also looks at current related work and addresses problems, limitations or gaps in the literature which serve as the baseline for the thesis work.

**3 Research Methodology** chapter outlines and justifies the research method used in the thesis, and provides a detailed description of the procedures and phases undertaken during the study.

**4 Tool Development** chapter details the software development method used along with an elaboration on its respective phases, justification of various choices, and a walk-through of the proposed solutions.

**5 Tool Evaluation** chapter concerns both evaluation phases from practitioners, detailing feedback on low-fidelity design, tool process flow, and final user interface design, to mention a few.

**6 Results** chapter include the results of the look at NFRs role as selection criteria, the scaling of the ICR framework to accommodate decision-making under uncertainty, and the overarching findings from the practitioner evaluation of the proposed tool.

**7 Discussion** chapter connects all the strings together to attempt to answer the research questions. Furthermore, any limitations and threats to validity are stated here.

**8 Conclusion** chapter serves as the final summary of the thesis and its findings, along with any future work.

# Chapter 2

# Background

This chapter details concepts and terminologies which are required for a smooth reading experience. The chapter consists of a combination of background and related works aiming at detailing any problems, gaps, or considerations which serve as a baseline for the thesis work.

The chapter begins with an introduction to software ecosystems, outlining the key entities and roles, and exploring the challenges inherent in such structures. It then looks into the concept of software vendor selection, emphasizing the role of non-functional requirements (NFRs) as critical selection criteria. It further explores various selection techniques, identifies gaps in current methodologies, and introduces a state-of-the-art (SOTA) framework relevant to this context. The chapter continues by reviewing current tool support and visualization techniques that are used within decision-making, before wrapping up the chapter in a summary of key takeaways that establish the research motivation.

## 2.1 Software Ecosystems

The concept of a software ecosystem, as described by Rani et al. [2], involves the decentralized development of software, where companies may opt to outsource their product, either entirely or partially, to one or more vendors. This term encompasses not only the participating companies but also the dynamic interactions among all involved parties, reflecting a collective investment in the software product's development and success [2, 3]. In Section 1.1, the question of; *which vendor should the keystone companies or ecosystem as a whole choose to integrate into the ecosystem, and how can the best decision be reached*, was introduced. This question was raised due to the multi-vendor ecosystems' complexities where coordination challenges and potential conflicts among numerous participants can further complicate decision-making. This question highlights one of the core challenges of software ecosystems, which will be further analyzed in Section 2.2.

Understanding the complexities of these ecosystems and making informed decisions about vendor integration necessitates a clear grasp of the various roles and

entities within these environments. To facilitate a deeper understanding and to ensure clarity throughout this thesis, it is essential, before exploring software vendor selection and the associated techniques, to define two key entities: *keystone company* and *vendor*, along with two key roles: *decision-maker* and *stakeholder*. The descriptions that follow will establish a shared vocabulary, crucial for effectively discussing and understanding the dynamics of software ecosystems:

***Keystone company***

The company that serves as the ecosystem owner, orchestrator, platform provider, or outsourcing company [2, 19]. In the context of this thesis, this is the entity which is responsible for and handles defining the criteria of which the vendors will be evaluated against and handling the vendor selection on behalf of the ecosystem.

***Vendor***

Vendors are the niche players or service providers of the ecosystem [5, 19]. These are the entities proposing a solution to the outsourced task aiming at being selected and included as a part of the ecosystem.

***Decision-maker***

A decision-maker is someone with trust and extensive knowledge about the company or ecosystem and its architecture, demonstrating high proficiency [2, 5, 20]. In this thesis, this is the role that evaluates the criteria importance and vendor capabilities on behalf of the keystone company.

***Stakeholder***

Stakeholders encompass business units such as the company board, sales, marketing, and so forth [19]. For the system proposed in this thesis, stakeholders are the ones responsible for project setup and selection criteria selection and definition, as well as serving as the final decisive power of which vendor to choose based on the decision-makers' and system's suggestion.

*Problem: Choosing the optimal vendor for integration into the ecosystem is a complex task, particularly in multi-vendor ecosystems where coordination challenges and potential conflicts among numerous participants can further complicate decision-making.*

## 2.2 Software Vendor Selection and Techniques

Due to the concerns among keystone companies within software ecosystems regarding the vendor selection choice and how they can reach the best solution, a look into what software vendor selection is, why this selection is done, why it is so important to get right, and which constraints decision-makers face in their pro-

cesses is needed. Therefore, in this section, the focus will be on addressing these questions to better understand which methods can be utilized and what can be done to assist these companies, ecosystems, and decision-makers in their pursuit of the optimal decision.

### 2.2.1 Software Vendor Selection

One of the key activities of the keystone companies in software ecosystems is the software vendor selection process. The process of software vendor selection in general involves identifying and ultimately choosing the vendor best suited to meet the specific goals or needs of the company [1]. In a software ecosystem, the same process applies [1]; however, in a multi-vendor ecosystem, vendors must comply not only with the needs of the keystone company but also with those of the entire ecosystem, amplifying the intricacies of the process. This process is crucial to get correct to increase the chance for software product and ecosystem success, as making the wrong choice can ultimately lead to a damaged reputation among, for instance, stakeholders and customers, or in the worst case, ecosystem failure [1]. The complexities and risk involved in this selection, along with the complex interactions among entities within software ecosystems [8] present barriers to adoption [21]. Moreover, the lack of effective modeling techniques further complicates the conceptualization process for participants and new adopters [22], despite the increasing popularity among software experts [5, 21]. This is further discussed in Section 2.2.3. This raises the questions of; how is this vendor selection done, who are responsible for the task and why is the selection process so prone to mistakes? As addressed in Section 2.2.1, the choice of which vendors to choose is done by the keystone company's decision-makers [1]. However, the decision-making process in these ecosystems is complex, ad-hoc, and manual, which adds additional layers of complexity to the system and presents a significant barrier to ecosystem adoption [5]. Even with dedicated personnel handling decision-making roles, the process remains subjective [1, 9], going on behalf of transparency [1]. To address these challenges and before looking into various proposed solutions or assisting frameworks and guidelines, it is crucial to establish a foundational understanding of the two central information sources essential for software vendor selection: *Request for Proposals* and *Vendor Proposals*. These documents play a vital role in structuring the selection process, ensuring that the complexities and subjectivity of decision-making can be navigated more effectively. Software vendor selection relies on specific information, which is divided into these two documents [1]. Additionally, the selection criteria serve as key elements, which will be further elaborated on in Section 2.2.2. The two documents that are essential during these processes are:

> ***Request for Proposals (RFPs)*** An RFP is a textual document issued by an organization to invite bids from vendors/contractors for the completion of a specific project. It publicly outlines the project details, objectives, and the outsourcing organization/ecosystem, along with the procedures for submit-

ting bids and the terms of the contract. [1][2]

***Vendor Proposals (VPs)*** A VP in software vendor selection is a textual document provided by a software vendor that details how their solution meets the specific needs outlined in the RFP. It includes information on the software's capabilities, technical specifications, implementation strategy, support services, pricing, and relevant projects to demonstrate the vendor's expertise and reliability, to mention a few.

It is worth noting that similarly to the terms in software ecosystems, these documents do neither have a shared consensus regarding their names as RFPs can also in some cases be referred to as *tenders* [23] or *supplier proposals*, while VPs can be discussed as *quotations* [23]. This further complicates the search for related literature, ensuring no variants are overlooked.

Now that both of these key documents have been introduced, looking at their role and why they are such important during the vendor selection process is a logical next step. The RFP and VPs are critical in the software vendor selection process because they facilitate a structured comparison of how different vendors plan to meet a company's strategic and architectural goals, a complex task requiring the processing and evaluation of large quantities of qualitative data [24, 25]. Given the subjective nature of this evaluation and the presence of multiple decision-makers with varying perspectives, these documents help standardize criteria and streamline decision-making, reducing potential conflicts and ensuring alignment with overall strategic objectives [1].

### 2.2.2 Selection Criteria and NFRs Role

Having established the fundamentals of software vendor selection within the given context, complete with detailed discussions on Requests for Proposals and Vendor Proposals, the next step involves looking at non-functional requirements (NFRs) and their significance as selection criteria in this context. Before looking at the SLR that was conducted on this topic and its findings (Further detailed in Section 6.1.3), introducing selection criteria and their role in the given context will be done. Furthermore, briefly touching upon various types of selection criteria, before lastly looking at NFRs role as selection criteria in software vendor selection in software ecosystems is deemed important.

In Section 2.2.1, the role of RFPs in the decision-making process was explored. Selection criteria play a key role in this context; they are either made to guide the creation of RFPs or derived directly from the RFPs themselves, acting as standards or benchmarks for evaluating and selecting various software vendors. These criteria are essential as they enable decision-makers to assess diverse aspects of the software vendors themselves and the software offered by vendors, including technical capabilities, cost, vendor reputation, and alignment with business objectives,

[1] https://www.investopedia.com/terms/r/request-for-proposal.asp

[2] https://www.techtarget.com/searchitchannel/definition/request-for-proposal

among others. Their importance is further emphasized by Lima et al. [26] stating that selection criteria are critical success factors, capable of improving relationships and reducing costs.

Moreover, advancing beyond mere textual analysis of RFP documents, a structured approach in using these criteria minimizes the risk of subjective or biased decisions, enhancing transparency and fairness in the selection process. This method enriches decision-makers' knowledge base, streamlining the evaluation process by considering the importance of different aspects, including functional requirements, non-functional requirements, architectural requirements, operational requirements, and more. Ultimately, these criteria are designed to be tangible, measurable, and actionable, ensuring that all software and vendor selections are precisely aligned with the strategic objectives and operational needs of the ecosystem, thereby supporting informed and effective decision-making.

The complexity of vendor selection within software ecosystems and the unique characteristics of each ecosystem further underscore the importance of these criteria. They serve as critical success factors that not only enhance relationships and reduce costs [26] but also enable decision-makers to manage the complexities of NFRs [10], which are often overlooked due to their subjective nature [9]. Interviews conducted by Olsson et al. [27] with key stakeholders underscore the need for precise criteria to effectively manage quality requirements. Additionally, the potential irreversible impacts of poor architectural decisions highlight the necessity for systematic evaluation [28], while the importance of incorporating architectural design decisions into software documentation reflects the need for clear, reusable criteria [9].

In conclusion, the literature suggests that well-defined selection criteria are essential for informed and successful decision-making. Although overly rigid criteria could potentially restrict the expertise of decision-makers, a more detailed breakdown of the criteria into sub-criteria and explanations could provide greater clarity and enhance the decision-making process [29].

*Gap: Literature emphasizes that well-defined selection criteria are crucial for informed and successful decision-making. However, the subjective nature of NFRs often leads to their neglect, creating a gap in understanding their role and interpretation in these processes, which could otherwise aid decision-makers more effectively.*

### 2.2.3 Multi-Criteria Decision Making Methods and SOTA Framework

Having laid the groundwork on software vendor selection within the given context, which includes details on Requests for Proposals, Vendor Proposals, selection criteria, and NFRs, the next step is to explore various decision-making methods and a state-of-the-art framework. The literature offers a variety of methods for vendor selection designed to tackle the complexities of decision-making within dynamic ecosystems. These methods are not only theoretical but have also been practically applied during the choice of appropriate software suppliers, encouraging

further exploration into robust decision-making methodologies. Importantly, the literature underscores the importance of an integrated approach that effectively combines the complexities of service selection and supplier evaluation contexts, as emphasized in Manikas' proposal [8].

Building on these insights, Multi-Criteria Decision Making (MCDM) methods emerge as particularly effective tools within this context. Techniques such as TOPSIS, ANP, and AHP are highlighted in the literature for their efficacy in assisting decision-making processes. These methods, which are critical for evaluating multiple conflicting criteria, have demonstrated their effectiveness in streamlining complex decisions in software vendor selection, as will be discussed next.

### TOPSIS - Technique for Order of Preference by Similarity to Ideal Solution

*TOPSIS*, short for *Technique for Order of Preference by Similarity to Ideal Solution*, is a multi-criteria decision analysis method that was developed through the 1980s and early 90's [30]. It is a method used to make complex decisions by evaluating alternatives against an ideal solution. It works by determining which option is closest to the best possible scenario (the ideal solution) and furthest from the worst (the negative-ideal solution) [30, 31]. While this approach effectively handles multiple criteria, providing a clear ranking of alternatives based on their relative closeness to the ideal solution and its ability to rank the advantages and disadvantages of the evaluation objects, it suffers from some weaknesses. More specifically the constraints of TOPSIS do not fit the reality as it uses crisp numeric values and lacks consistency and reliability checks [32].

### ANP - Analytic Network Process

The *Analytic Network Process (ANP)* is another MCDM method often used for supplier selection. ANP is great for dealing with complex decisions because it handles dependencies among decision factors well [33], uses different types of data, and provides thorough evaluations. However, its flexible and deep analysis comes with downsides. The network's complex structure, which includes feedback and interdependencies, makes it tough for decision-makers. They have to consider many elements that may affect each other, leading to potential judgment mistakes, sensitivity to changes in input, and challenges in finding and fixing errors due to no standard way to check consistency [34].

### AHP - Analytic Hierarchy Process

*AHP*, short for *Analytic Hierarchy Process* as presented by Saaty [35] is the last MCDM method under review in this study. AHP, like the other two, helps to simplify and accelerate the natural process of decision-making [36]. As for all these methods, it was presented as a way to address complex decision-making scenarios where multiple criteria are involved. AHP breaks down the decisions into a hierarchy of interrelated elements (criteria, sub-criteria, alternatives, etc.) [35]. These

**Table 2.1:** Criteria-Criteria Matrix

| | Usability | Reliability | Security |
|---|---|---|---|
| Usability | 1 | 2 | 5 |
| Reliability | 0.50 | 1 | 7 |
| Security | 0.20 | 0.14 | 1 |

**Table 2.2:** Vendor Comparison - Usability

| **Usability** | Vendor X | Vendor Y |
|---|---|---|
| Vendor X | 1 | 0.20 |
| Vendor Y | 5 | 1 |

elements are thereafter placed in a pairwise comparison matrix to determine their relative priorities, usually done through a scoring from 1 to 9 (See table 2.5) [35]. At this level, decision-makers/evaluators assign scores to all the different combinations to fill the matrix. For vendor selection, this is done both *criteria-criteria* wise and *vendor capability-criteria* wise [1] as illustrated in Tables 2.1, 2.2, 2.3, and 2.4. Lastly, a consistency check is done to assess the reliability of the comparisons made.

Having established the baseline of the AHP, it is insightful to explore some of its practical applications, particularly in vendor and supplier selection. AHP has demonstrated its robustness in various decision-making settings:

Hruška et al. [36] developed a model using the AHP for supplier selection and applied it within a manufacturing company to evaluate three potential suppliers. Their results underscored the model's effectiveness in helping decision-makers to systematically assess the strengths and weaknesses of suppliers according to predefined criteria.

In the telecommunications sector, Tam et al. [37] employed the AHP to create a model that was tested in a real-world scenario for vendor selection. This model proved effective in improving group decision-making and simplifying the vendor selection process by structuring decision-making, fostering consensus among stakeholders, and adapting to changing business needs. Nevertheless, challenges related to data collection and computation can become apparent as the number of criteria and vendors grows.

Tahriri et al. [38] also applied the AHP model but in a steel manufacturing company to evaluate and select suppliers. The practical application of the AHP model highlighted its effectiveness in facilitating the supplier selection pro-

**Table 2.3:** Vendor Comparison - Reliability

| **Reliability** | Vendor X | Vendor Y |
|---|---|---|
| Vendor X | 1 | 0.14 |
| Vendor Y | 7 | 1 |

**Table 2.4:** Vendor Comparison - Security

| **Security** | Vendor X | Vendor Y |
|---|---|---|
| Vendor X | 1 | 3 |
| Vendor Y | 0.33 | 1 |

**Table 2.5:** The Fundamental Scale As Presented by Saaty [35]

| **Importance** | **Definition** | **Explanation** |
|---|---|---|
| 1 | Equal importance | Two activities contribute equally to the objective |
| 3 | Moderate importance of one over another | Experience and judgment slightly favor one activity over another |
| 5 | Essential or strong importance | Experience and judgment strongly favor one activity over another |
| 7 | Very strong importance | An activity is favored very strongly over another; its dominance demonstrated in practice |
| 9 | Extreme importance | The evidence favoring one activity over another is of the highest possible order of affirmation |
| 2, 4, 6, 8 | Intermediate values | Represent compromise between the priorities above |

cess. By methodically evaluating various criteria, the model significantly enhanced decision-making, enabling the identification of optimal suppliers for the steel manufacturing company. This successful application reaffirmed the AHP's capacity to navigate complex decision-making environments, thereby improving both the efficiency and effectiveness of purchasing processes.

Despite its effectiveness in complex decision-making processes, as demonstrated in Section 2.2.3, the AHP and its alternatives are not without limitations. These include the subjectivity of judgments, scalability issues concerning both criteria and the number of decision-makers, assumptions of criteria independence, and a lack of clarity on the root causes of inconsistencies. Furthermore, conflicts among decision-makers often emerge only at the later stages of the process [1]. To address the latter two issues, Rani et al. [1] introduced a framework that utilizes the Inconsistency and Conflict (ICR) method in conjunction with AHP for vendor proposal analysis and selection in software ecosystems (See Figure 2.1). This framework represents a significant advancement in the field, positioning it as an ideal subject for further exploration in this thesis. While it enhances the adaptability of AHP to these complex scenarios and addresses some of its inherent limitations, it does not fully mitigate the challenge of decision-making under uncertainty — a topic that will be elaborated in Section 2.2.3. Moreover, the current implementation requires familiarity with the code base, restricting its accessibility to only those proficient with its technical aspects. This limitation often compels decision-makers to stick to their ad-hoc processes [1, 5].

*Gap: Although the ICR framework enhances the adaptability of AHP by addressing*

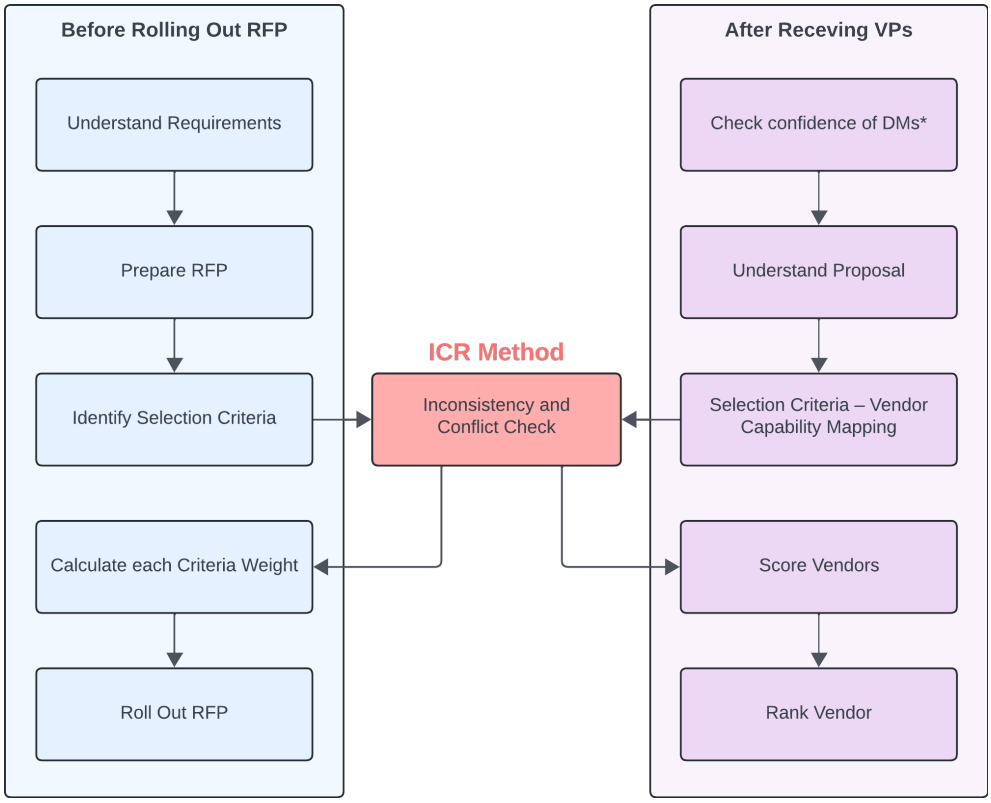


**Figure 2.1:** ICR Framework Proposed by Rani et al. [1]

*some of its inherent limitations, it faces challenges with integration into existing company processes. Currently, the requirement for code familiarity limits its use to individuals with technical expertise, leading decision-makers to rely on ad-hoc methods. To improve both adaptability and integration, the framework should incorporate a user interface and a logical layer that bridges the gap between the framework's algorithm and its end-users.*

### Fuzzy Set Theory

Another issue related to AHP in general and in the proposed framework by Rani et al. [1] is the lack of consideration of decision-making under uncertainty. To address the challenge posed by imprecise and uncertain data within the AHP framework, the adoption of a practical strategy, incorporating a linguistic scale of judgment and fuzzy numbers for performance assessment [39] are often present. The integration of AHP with Fuzzy Set Theory (FST) has gained traction, with hybrid AHP/FST models (often referred to as *Fuzzy Analytic Hierarchy Process (FAHP)*) proving successful in various software and supplier selection studies, effectively leveraging the strengths of both methodologies while mitigating their respective weaknesses [40–48]. As the inclusion of FST with AHP enhances the decision-

making methodology's adaptability and usability as proven in diverse case studies [46, 49, 50], looking at how such a concept can be used to scale the ICR framework by Rani and colleagues [1] to accommodate for decision-making under uncertainty is central. The success of FST as a complement to the AHP methodology is further substantiated by findings from the SLR conducted for this thesis. Out of the 31 papers that underwent review, five explicitly discuss the effective application of weights to selection criteria using FST as their foundational approach [51–55].

Building on this foundation, a step-by-step guide on how to utilize Fuzzy AHP (FAHP) in supplier selection is presented by Ayhan [56]. This guide has been instrumental in integrating the fuzzy algorithm into the ICR framework and tool, as further detailed in Section 4.4.4. FST's overarching concept allows for partial membership, enabling elements to belong to a set to varying degrees. This is visually represented in Figure 2.2 and differs significantly from traditional set theory approaches like AHP, which utilize crisp sets (Saaty Scale), as seen in Table 2.5. Furthermore, FST supports textual importance levels, which resonate more naturally with human cognitive processes, thereby reducing cognitive load for decision-makers.

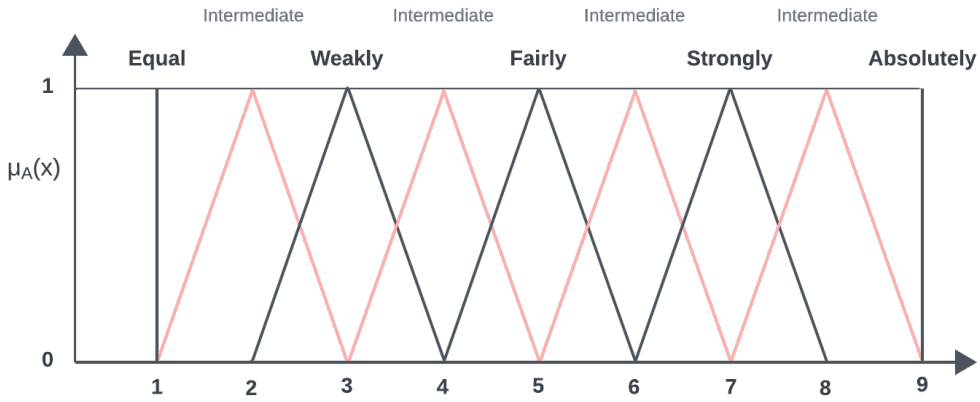


**Figure 2.2:** Fuzzy Triangulation

*Gap: Although the ICR framework enhances AHP's adaptability, it still struggles with decision-making under uncertainty. Traditional AHP cannot fully capture subjective human judgments when decision parameters are ambiguous. Integrating fuzzy logic into the ICR framework could improve its capability to handle uncertain and imprecise information, thus aiding decision-makers in complex scenarios.*

## 2.3 Tool Support and Visualization Techniques

So far, the background has introduced software ecosystems, software vendor selection, and various associated techniques. It has been identified that there is a press-

**Table 2.6:** Saaty Scale - Fuzzy Triangular Scale Mapping

| Saaty Scale | Definitions | Fuzzy Triangular Scale |
|---|---|---|
| 1 | Equally important | (1, 1, 1) |
| 3 | Weakly important | (2, 3, 4) |
| 5 | Fairly important | (4, 5, 6) |
| 7 | Strongly important | (6, 7, 8) |
| 9 | Absolutely important | (9, 9, 9) |
| 2 | | (1, 2, 3) |
| 4 | The intermittent values | (3, 4, 5) |
| 6 | between two adjacent scales | (5, 6, 7) |
| 8 | | (7, 8, 9) |

ing need for a tool that streamlines the decision-making process. Consequently, it is crucial to explore existing tools and various visualization techniques to extract insights that could inform the development of such a tool.

To enhance the ICR framework's adaptability and facilitate seamless integration into existing company processes, the framework should incorporate a user interface and logical layer that bridges the algorithm and end-users. This enhancement, as also wished for by the framework developers [1], aims to deliver a user-friendly experience for practitioners and improve the efficiency and reliability of the decision-making process, as emphasized by the importance of visualizations in supporting decision-makers [57].

To accommodate this need, a semi-structured literature review was conducted to assess the current methodologies, frameworks, and tools to understand the support available to decision-makers and to determine the functionality and appearance of such tools. The common finding is that there is a limited priority on the user interfaces and tools utilized, but rather a focus on the methods and processes. A handful of studies talk about the usage of MS Excel, such as in the works of Gencer et al. [58], Kilincci et al. [49] and Dargi et al. [59], however, the focus on and details of other tools prove limited.

As research on existing tools proved limited, trying to understand the role of visualizations in decision-making was deemed important. While the current state of the literature on visualizations for software vendor selection remains limited, some studies motivate the efficiency of visualizations in decision-making in general. Killen et al. [57] looked into how decision-makers in project portfolio management leverage visualizations to enhance cognitive processes and improve decision-making. In their study, a positive relationship between the decision-makers' use of visualizations and project portfolio success was identified, which is mediated by decision-making success. While they show this positive relationship, they also emphasize the significance of cognitive factors like familiarity with visualizations and the tendency to use heuristics (mental shortcuts) in decision-making. Additionally, the authors draw attention to the fact that visualizations facilitate the recognition of patterns and trends that might not be easily visible in raw

data formats and that more informed decision-making based on a deeper understanding of the data may be achieved using visualizations. However, the paper also highlights the importance of avoiding overwhelming decision-makers in these decision-making tools. Finding a balance here proves to be key for decision-making success as visualizations can provide a more objective view of the data, helping to avoid biases and errors that can arise from heuristic thinking [57]. The need for visualizations during decision-making is also underscored by Rani et al. [1], particularly in software vendor selection. They state that company stakeholders and decision-makers may require a mechanism to effectively represent and visualize vendor information without going into more detail.

When seeing the upsides of the use of visualizations in decision-making, understanding the drawbacks is just as important. The studies by Killen et al. [57] and Padilla et al. [60] suggest that well-designed visualizations can enhance decision-making by aligning with the user's mental schemas and reducing cognitive load, thereby improving accuracy and efficiency. However, they also warn that poorly designed visualizations can lead to misinterpretations and biases, particularly if the visualizations require specific domain knowledge or if users apply their own biases to the data.

Overall, the effectiveness of visualizations in decision-making heavily depends on their design and the user's prior knowledge and experience, underscoring the need for careful design and implementation to avoid cognitive overload and potential errors.

*Considerations: While visualizations have proved beneficial in assisting decision-makers make more informed and successful decisions, the effectiveness of visualizations in decision-making heavily depends on their design and the user's prior knowledge and experience, underscoring the need for careful design and implementation to avoid cognitive overload and potential errors.*

## 2.4 Key Takeaways

Throughout this background chapter, problems, gaps and considerations have been addressed. In this section, before recommending further reading, these points will be summarized as key takeaways.

**Problem:** The selection of the optimal vendor for integration into a multi-vendor ecosystem presents a complex challenge. This complexity is further amplified by coordination difficulties and potential conflicts among numerous participants, complicating the decision-making process.

**First Gap:** While the literature stresses the importance of well-defined selection criteria for informed and successful decision-making, the subjective nature of Non-Functional Requirements (NFRs) often leads to their neglect. This oversight creates a significant gap in understanding their role and interpretation, which could otherwise enhance decision-making efficacy.

**Second Gap:** The ICR framework, although it improves the adaptability of the AHP by mitigating some of its inherent limitations, still faces challenges in integrating into existing company processes. The current need for code familiarity restricts its usability to those with technical expertise. This limitation forces decision-makers to resort to ad-hoc methods. To resolve this, the framework should include a user interface and a logical layer that connects the algorithm directly to the end-users, enhancing both adaptability and ease of integration.

**Third Gap:** Furthermore, the ICR framework, despite enhancing the adaptability of AHP, struggles with decision-making under uncertainty. Traditional AHP fails to fully capture the nuances of subjective human judgments when decision parameters are ambiguous. Integrating fuzzy logic could significantly enhance the framework's ability to manage uncertain and imprecise information, supporting decision-makers in complex scenarios.

**Considerations:** Although visualizations have proven beneficial in assisting decision-makers to make more informed and successful decisions, their effectiveness depends heavily on their design and the user's prior knowledge and experience. This underscores the necessity for careful design and implementation to avoid overwhelming the decision-makers and prevent potential errors.

# Chapter 3

# Research Methodology

This chapter outlines and justifies the research method used in the thesis, and provides a detailed description of the procedures and respective phases undertaken during the study.

## 3.1 Action Research

To properly be able to address and evaluate the research questions defined in Section 1.4, an overarching scientific approach is needed to maintain the integrity and validity of the findings. As the thesis focuses on solving issues faced by practitioners, specifically decision-makers in their software vendor selection processes, *Action Research* is chosen as the method of choice, as further addressed in the upcoming sections. Action research is a research methodology that is designed to offer support in social structures, such as company practices, processes, and procedures, that generally are vastly demanding to evaluate, due to interference that is only partly understood and hard to control [61]. Contrasting with Technology Research, which primarily focus on technological innovations, ideally aiming at the development of artifacts, action research focuses on the active engagement with stakeholders to identify and address practical challenges [61] they meet in their industry practices, making it a perfect research methodology for this thesis. Solheim et al. [62] propose a step-wise, cyclical approach for action research consisting of *Problem Analysis*, *Innovation*, and *Evaluation*, which is further delineated into the five phases as presented by Susman et al. [63]; *Diagnose*, *Plan*, *Act*, *Evaluate*, and *Reflect*, as depicted in Figure 3.1 [61]. Through these phases, a problem area and the need for improvement are identified, actions for solving or remedying the problem are planned and performed, and lastly, the effects are analyzed and evaluated regardless of the outcome [61].

In each of the upcoming sections, the thesis presents action research in its specific context. However, before making these connections, it is necessary to deepen the understanding of the stages involved in action research. Figure 3.1 illustrates the stages of the action research process. The stages and explanations introduced next originate from the work of Stølen et al. [61].

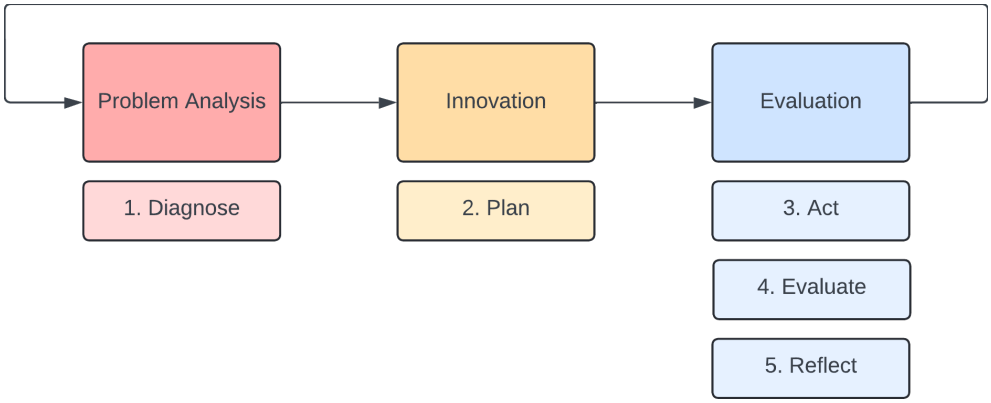


**Figure 3.1:** Action Research Methodology

## 3.2 Problem Analysis

The *diagnose* phase within the overarching problem analysis phase revolves around the identification of a common understanding of the problem to be solved. This can be obtained through interactions with practitioners who raise a need or frustration with current processes or artifacts, or preliminary literature reviews from social structures. The result is a description of the need and problem area of the organization.

### 3.2.1 Problem Analysis in the Context of the Thesis

For the problem analysis phase of action research, three steps served central in this thesis; *Identifying the problem (1)*, *Defining the problem (2)*, and *Understanding the problem context (3)*. Such a thorough problem analysis ensures that the actions taken are informed and targeted effectively, enabling an effective understanding of the needs in the social structures [61]. These steps are highlighted throughout the rest of the section.

The problem analysis of the thesis has been multifold. Upon getting introduced to the topics of software vendor selection in software ecosystems, interest in how the processes and constraints of practitioners could be improved and solved arose. The process of understanding the current literature landscape of software vendor selection processes in software ecosystems begun, to identify gaps that can be filled.

The literature findings are addressed in detail in Section 2 while the distribution of the literature search, the reasoning why the respective strategies were chosen, and a summary of the problem analysis is addressed here. As detailed in the upcoming paragraphs and illustrated in Figure 3.2, three overarching tasks were conducted for problem analysis.

Initially, an overarching problem identification within software vendor selection was conducted to understand why software vendor selection is deemed so

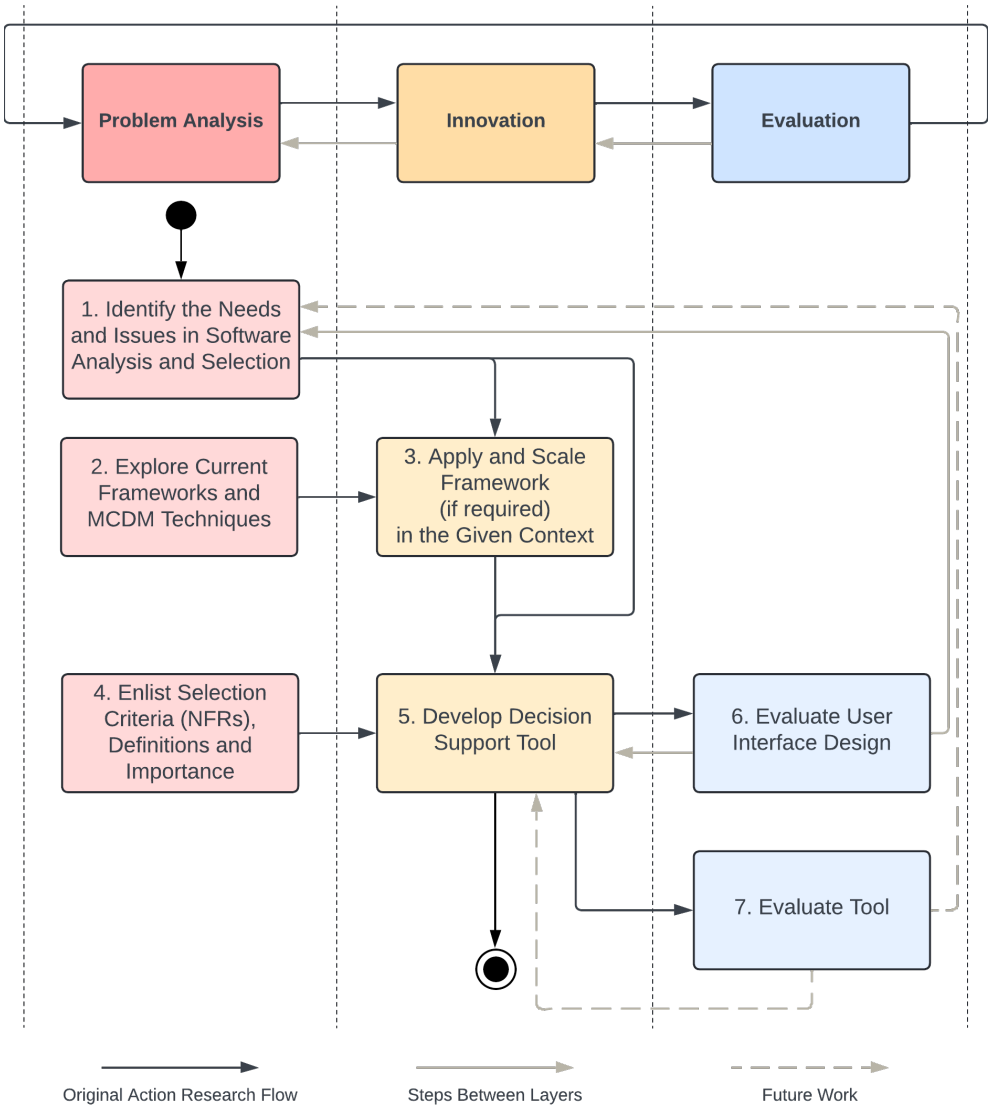


**Figure 3.2:** Thesis Research Method

difficult (1). Utilizing a semi-structured literature review allowed for exploration of multiple topics and further identifying and exploring gaps that arose. From the review, it became evident that current methods for assisting decision-makers in their practices lack adaptability and usability prioritization (2.1) and the intricacies of NFRs make them difficult to use as selection criteria (2.2). Most of the prioritization is directed towards proposing methods and frameworks, and some requirements, which only get you a part of the way (3).

Upon some of the findings from the preliminary literature review, another semi-structured literature review was conducted. Here the goal was to identify limitations in existing frameworks and tools for decision-making, study the role of visualizations in decision-making and such tools, and the respective cognitive challenges of using visualizations in such tools (1) to further elaborate on the preliminary findings. The literature review findings indicated the need for a tool that streamlines the decision-making process (2) as there is minimal talk about the use of visualizations and tools to assist software vendor selection (3). Even despite studies showing that there is a positive relationship between users' use of visualizations and decision-making success.

The last, and most comprehensive review of the literature was conducted through an SLR. The overarching goal of the literature review was it being an SLR of NFRs' role as selection criteria in software vendor selection in software ecosystems. Conducting an SLR to look at this was done as it would provide a comprehensive and in-depth understanding of the current literature's perspective on how NFRs can be effectively utilized and defined within an SECO context. The SLR was further divided into two problem analysis phases in the context of the thesis. Initially, a comprehensive understanding of the existing research landscape related to software ecosystems, particularly concerning the consideration of NFRs, was carried out through a dedicated search for SLRs in the domain of software outsourcing. There it was found that despite the existence of several SLRs addressing software ecosystems and software outsourcing, a limited number of these have explored NFRs in general, but none of them have explored NFRs as selection criteria within software ecosystems (1). The evaluation of such SLRs during the literature review was conducted as there was no reason conducting an SLR on that topic if a similar one already existed. This lack of focus on NFRs as selection criteria (2) despite its importance (as addressed in Section 2.2.2) served central in the problem definition phase. Lastly, the second side of the problem analysis was reviewing the primary sources that served as a result of the SLR methodology (Section 6.1). Here a broader understanding of NFRs role as selection criteria within this context (3) was identified.

## 3.3 Innovation

Upon the completion of the problem analysis phase, the next phase in action research is innovation. This phase involves *planning* the actions necessary to address the identified issues to remedy the situation [61]. Here the overarching hypothesis

is that by executing the planned actions, the problem will be reduced or completely eliminated. This requires an identified and clearly defined problem, along with an understanding of the problem context, as action research aims at solving real-world problems through practical solutions. One of the strengths of innovation in action research is its collaborative approach involving researchers and practitioners working together. This collaboration not only facilitates the gathering of diverse ideas and expertise which leads to more creative and innovative solutions, but also ensures that the solutions are informed by the real-world experiences and feedback of those affected by the problem. When combined with the iterative cycles of action research, this innovative approach promotes continuous refinement and adaptation of ideas. This encouragement of experimentation and adjustment is crucial for fostering innovation.

### 3.3.1 Innovation in the Context of the Thesis

The innovation phase in this thesis' work began upon the completion of the problem analysis stage. This stage involved planning actions necessary to address the issues identified during the problem analysis stage. Two separate processes were initiated with the overarching goal of forming a complete solution. This separation was done as the tasks could be worked on and completed in isolation, before merging together to a final solution at the end, and was assessed as the most effective way of solving the tasks. In this way, if one task went to a halt, the two other tasks could still progress.

One of the latest and state-of-the-art frameworks lacked the consideration of decision-making under uncertainty and required to be made more adaptable and usable. To address these considerations, the framework is scaled to utilize Fuzzy Set Theory (Detailed in Section 6.2) and propose a web tool (Detailed in Section 4) bringing the decision-makers through their processes, bridging the framework logic and the user.

Throughout the innovation phase, the availability of practitioners was limited. However, to accommodate this, continuous discussions were held approximately once a week with the framework developers to gather feedback and new ideas. This was particularly beneficial during the innovation phase due to their extensive experience and knowledge about the state of the art and the state of the practice within the field.

## 3.4 Evaluation Strategies

Once the problem analysis is finished and the innovation phase is executed, the last of the three pillars of action research begins; *evaluation*. Susman et al. [63] divide the evaluation phase into three distinct phases, as detailed next. Initially, the *act/implement* phase puts the planned actions into life to obtain the desired change. This may involve the development of new technologies or artifacts, or

altering existing workflows, among other things. Following this, the (sub) *evaluation* phase assesses whether these actions achieved the intended outcomes, and evaluates whether these outcomes effectively address the initial problems. The concluding phase, *reflection*, documents and reflects on the learning outcomes for all stakeholder groups; the organization, action researchers, and the international research community.

Evaluation methods vary widely, but they all aim to determine if the actions have achieved the intended effects and addressed the problems. McGrath [64] proposes eight evaluation strategies distributed across four categories, suggesting a triangulation approach to *generality*, *precision*, and *realism*. However, accommodating all these aspects is often unfeasible [62]. Researchers thus select strategies that best balance these properties and provide the most value to the evaluated artifact [61]. According to Stølen et al. [61], the most fitting strategies for action research include *field experiments*, *surveys*, and *interviews*.

These factors informed the selection of evaluation strategies for this thesis. The strategies considered and selected, along with the rationale for these choices, are discussed in the next section (Section 3.4), while a detailed description of the evaluation process is presented in Section 5.

### Evaluation in the Context of the Thesis

The evaluation process closely mirrored the iterations of action research. While some intermediate evaluation have been done with the framework developers to ensure that the methods and tool align with industry practices, two official evaluation rounds with expert practitioners was conducted throughout the research work (Chapter 5).

While semi-structured qualitative interviews have limitations of being time-consuming, limited to scalability and the data becoming highly subjective, the depth of the data serving nuanced insights through its flexibility makes it a perfect evaluation method to acquire practitioner insights and needs. The use of such a method offer the opportunity to gather detailed feedback and understand the context in which the tool is used, including any unique or unexpected ways it impacts decision-making processes.

Despite surveys' providing quantifiable data that can be scaled and distributed to a bunch of users, its limitations with respect to capturing the depth of user experiences and its rigid structure limits the ability to explore unexpected topics that might arise during an interview.

Several evaluation strategies were considered when deciding upon the evaluation strategy for the thesis. For the evaluation of the low-fidelity user-interface design and tool process flow, semi-structured, qualitative interviews were planned for and conducted. For the final tool evaluation, the initial idea was conducting workshops or participant observations with a set of practitioners. Such an evaluation method for the tool and its process flow would offer valuable, in-depth insights into how users use the tool. This proved to be unfeasible, however, as

development ended up taking more time than initially planned for, and more importantly, both the availability of expert practitioners and economical constraints. Providing the practitioners with a survey was also considered to accommodate practitioner availability, but its limitations for tool evaluation resulted in the idea being retired. This resulted in a new round of semi-structured, qualitative interviews as this proved successful in the first round of evaluation, however, this time with more practitioners (Section 5.2). Additionally, it did not require a gathering of practitioners, nor a budget.

## 3.5 Research Ethics, Data Collection, Analysis and Tools

As interviews were chosen as the evaluation strategy of the thesis, looking at research ethics, the data collection and analysis process and how practitioners were chosen is deemed important.

The interviews were conducted to assess and validate the proposed tool. Four interviews were conducted over two iterations with three practitioners; one interview for low-fidelity design evaluation with one of the practitioners and three interviews with the three practitioners for the final tool evaluation. The interviews were semi-structured and guided by an interview guide (one per iteration) which can be seen in Appendix C. These interviews were conducted under the ethical consideration of the Norwegian Agency for Shared Services in Education and Research (SIKT), ensuring considerations such as all participants signed informed consent documents and being informed about their data and its processing. Furthermore, the interviews were recorded using Diktafon[1] and stored using Nettskjema[2], where the voice files were transcribed and made anonymous before being deleted. The transcribed data was in turn analyzed thematically utilizing NVivo[3] to code the interviews and ease the analysis process.

Lastly, regarding the choice of practitioners, the process of getting in touch with and acquiring the time of practitioners was challenging as will be further discussed in Section 7.2.1. Practitioners were acquired due to their experience in decision-making practices, which was deemed crucial given the nature of action research. A group of practitioners was reached out to over mail and LinkedIn messages. These were practitioners who already had shown interest in the research field through participation in similar studies. Three of the practitioners cleared time in their schedule to participate.

[1] https://www.uio.no/english/services/it/adm-services/nettskjema/help/nettskjema-dictaphone.html

[2] https://nettskjema.no/?lang=en

[3] https://lumivero.com/products/nvivo/

# Chapter 4

# Tool Development

In this chapter, the development of the tool is explored. The tool development covers the thesis innovation phase and involve both the *Apply and Scale Framework in the Given Context* and the *Develop Decision Support Tool* (Figure 4.1). The discussion begins by examining the software development method employed, followed by a detailed walk-through of each phase in the process. This includes the planning phase, where requirements elicitation, defining the project scope, and selecting appropriate tools and technologies are elaborated upon. Additional aspects of the development process are also considered, such as the use of version control. Subsequently, the design phase is introduced, featuring discussions on initial sketches, low-fidelity user interface designs, and the application of UML diagrams. The chapter concludes with the implementation phase, which covers the architecture design, an in-depth review of the user interface design and tool process flow, strategies for separation of duties and access control, and the integration of Fuzzy Set Theory.

## 4.1 Software Development Method

The Software Development Life Cycle (SDLC) served central throughout the tool development process. Various SDLC models exist, including popular ones like Waterfall, Iterative, Spiral, and Agile [65]. Methods like waterfall were deemed to have too rigid a structure for this project. The rigid structure of the Waterfall model made it unsuitable for this project. Instead, the Agile methodology[1], known for its flexibility and adaptability, was considered. This approach facilitates rapid adjustments and the iterative discarding of ideas, making it possible to meet product requirements through incremental development and timely modifications.

Given the scope, timeline, and exploratory style of this project, an Agile strategy appeared advantageous. However, the risk of *scope creep*[2], due to inexperience in project management and the specific vendor selection processes for which the tool

[1] https://agilemanifesto.org

[2] https://en.wikipedia.org/wiki/Scope_creep

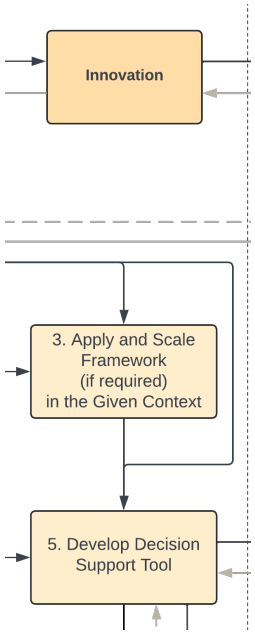


**Figure 4.1:** Action Research Methodology, Innovation Phase

was being developed, was a significant concern. The phases that did not provide sufficient value relative to the effort required to accommodate them also led to concerns about the suitability of a pure Agile approach. Therefore, a hybrid methodology was adopted, combining the structured foundation of Waterfall with the flexibility and maneuverability of Agile (Figure 4.2).

In this project, the SDLC process utilized the common phases as defined by Amazon Web Services [65], yet it diverged from a strict Waterfall approach by incorporating feedback loops across various stages. This revisiting of earlier stages, informed by insights from later stages, enhanced the quality and relevance of the tool's outputs. It also facilitates a learning environment where the development team can adapt based on the practical experiences and challenges encountered during implementation. This adaptability is particularly beneficial in complex projects where initial assumptions may require reevaluation, like in this one.

The key phases actively engaged in during this thesis' development process are highlighted in yellow in the model (Figure 4.2), specifically: *planning*, *designing*, and *implementation*. These phases are detailed in their respective sections, however, an initial overarching look at the role of the different phases is introduced here:

The *planning* phase included identifying the objective of the to-be-developed tool and making the initial key decisions. These decisions include requirements elicitation, project scope definition, and tool and technology selection, and are further discussed in Section 4.2. The decisions made during the planning phase directly influenced the efficiency and success of the design and implementation phases. This ensured that the project remained aligned with its initial objectives

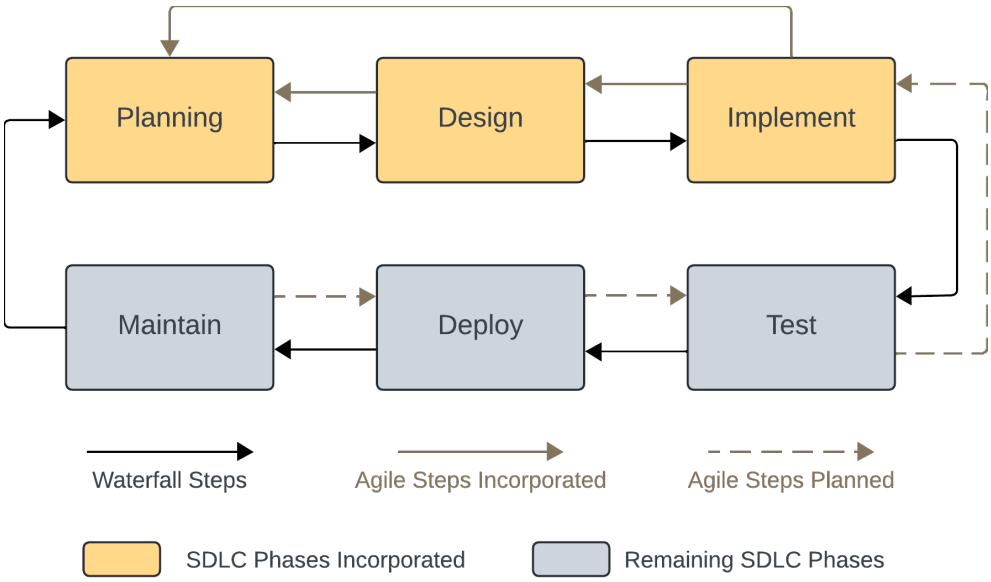


**Figure 4.2:** Software Development Method

and stakeholder expectations throughout the project period.

The *design* phase is further detailed in Section 4.3 and includes analyzing the requirements and developing a low-fidelity design based on the requirements. Furthermore, UML diagrams were created to better understand the entire process flow of the tool and user activities in detail to ease the development process.

Lastly, during the *implementation* phase, the previous phases and their findings were integrated into the implementation phase. At this phase, the main activity of developing the solution was done. The implementation, both looking at system architecture and final tool user interface design is further discussed in Section 4.4.

## 4.2 Planning Phase

As previously stated, the *planning* phase included identifying the objective of the to-be-developed tool and making the initial key decisions. These decisions include requirements elicitation, project scope definition, and tool and technology selection Furthermore, this phase involved the following key activities:

**Requirement Elicitation:** Detailed discussions with stakeholders, specifically the developers of the baseline framework, and literature analysis were done to understand and document the practitioners' needs and expectations. This helped in formulating a comprehensive list of functionalities that the software needed to provide, as further detailed in Section 4.2.1.

**Project Scope Definition:** Clearly defining what the project will and will not cover, thereby setting precise boundaries and deliverables for the project.

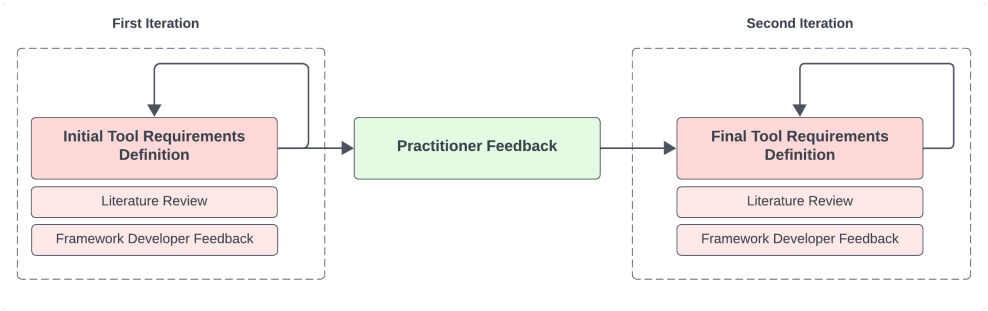


**Figure 4.3:** Requirement Definition Process

This is further detailed in Section 4.2.2.

**Tool and Technology Selection:** Deciding not only on programming languages but also on frameworks, development environments, and tools that align with the project goals and help in effectively reaching these goals. This is covered in Section 4.2.3.

### 4.2.1 Requirement Elicitation

The first phase of the planning phase was the requirement definition phase. The requirement definition phase followed an iterative process, incorporating intermediate feedback cycles. There were two main iterations of requirement definition: one for the initial set of requirements made for the low-fidelity user interface design and process flow, and another for the final tool development phase (Figure 4.3). These requirements were crucial for understanding user needs, ensuring that all aspects of the process flow were considered before development. The first requirements iteration was utilized and refined based on practitioner feedback and new understandings, serving as the foundation for the second requirement elicitation iteration serving as a baseline for the final prototype (See Figure 4.3).

The first step of the requirements engineering process was to formulate an objective for the tool that is to be developed. This objective was used as a baseline for the requirement elicitation, and ended up being:

*The tool should enhance and streamline the decision-making process and associated functionalities for practitioners by improving usability.*

To ensure important aspects of the process are considered, a set of specific tool requirements were listed. The initial requirements from the prototype can be seen in Appendix D. These were feature requirements and were not divided into any prioritization list. For the prototype implementation, however, the requirements were grouped using MoSCoW Analysis[3] as shown in Table 4.1.

The second requirement elicitation iteration consisted of the requirements listed in Table 4.2 and is listed in no particular order.

[3] https://monday.com/blog/project-management/moscow-prioritization-method/

**Table 4.1:** MoSCoW Analysis of Project Requirements

| | | |
|---|---|---|
| **M** | *Must-Have* | Essential requirements necessary for project success. |
| **S** | *Should-Have* | Important enhancements but not critical for launch. |
| **C** | *Could-Have* | Additional features beneficial if resources allow. |
| **W** | *Will-Not-Have* | Out of scope features not planned included in this phase. |

Additionally, a set of non-functional requirements were addressed as important during tool development. These were, in no particular order:

**Req 1** Usability
**Req 2** Security
**Req 3** Extensibility
**Req 4** Maintainability
**Req 5** Scalability

### 4.2.2 Project Scope Definition

Upon requirements being gathered and defined, the scoping of the project, more specifically the tool, followed. At this stage, the focus was on clearly addressing expectations and the overall aim of the tool, along with deadlines to avoid the scope creep risk as previously stated and have a clarification of expectations. The planning phase also allowed for mapping out which activities should be done within which deadlines to reach the overarching goal of the two main tool evaluation iterations; low-fidelity design evaluation and final tool evaluation (Figure 3).

The objective of the tool, detailed in Section 4.2.1, guided the evaluation of the development progress to determine when the tool had reached a satisfactory level. As outlined in Section 4.2.1, the requirements were organized according to the MoSCoW method, allowing for clear separation of what was mandatory, what ideally should be there, what is nice to have, and what would not be included. Only one requirement was changed to the 'will not have' category after the first iteration, but subsequent feedback from practitioners and new insights led to the addition of seven new requirements as shown in Table 4.2. The establishment and subsequent modifications of these initial objectives were conducted in collaboration with the framework developers. They also participated in the formal approval process for any scope changes, allowed by the agile maneuverability. While the project scope remained largely consistent throughout the project, it was slightly adjusted following discussions with practitioners.

### 4.2.3 Tool and Technology Selection

The last step in the planning phase was the tool and technology selection. This included the selection of programming languages, frameworks, development environments, and tools that help effectively reach the defined goals. First, a look at

**Table 4.2:** Tool Requirements

| Req # | M/S/C/W | Req | Status |
|---|---|---|---|
| 1 | **M** | Define and new project with stakeholders and decision-makers | *New* |
| 2 | **M** | Show Request for Proposals and Vendor Proposals during scoring | |
| 3 | **M** | Show selection criteria and explanations during scoring | |
| 4 | **M** | Let decision-makers assign scores to selection criteria as<br>**M** main-criteria<br>**S** sub-criteria<br>**S** questions | |
| 5 | **M** | Let decision-makers assign importance to criteria | |
| 6 | **M** | Let decision-makers assign scores to vendor capabilities | |
| 7 | **S** | Allow score and importance assignment using linguistics instead of crisp numeric values | *New* |
| 8 | **S** | Have a dashboard to show the current state of the decision-making process | *New* |
| 9 | **M** | Let stakeholders assign new criteria to the vendor selection process and modify explanations | |
| 10 | **M** | Highlight inconsistencies and conflicts in importance for the criteria | |
| 11 | **M** | Highlight inconsistencies and conflicts in importance for vendor capabilities | |
| 12 | **C** | Suggest actions to take based on conflicts | |
| 13 | **M** | Show final vendor rankings | |
| 14 | **S** | Visualize vendor data | |
| 15 | **W** | Show pair-wise matrix | *Changed* |
| 16 | **M** | Have user-based access control | *New* |
| 17 | **M** | Persistently store data | *New* |
| 18 | **C** | Chat or anonymous contact of vendors capabilities | *New* |
| 19 | **S** | Report generation | *New* |

the various programming languages and frameworks utilized to achieve the goals and a justification of the choice is deemed necessary.

**React**[4] is a popular JavaScript library for building single-page applications, combining HTML, JavaScript and CSS. React is one of the most popular web development frameworks out there[5], offering comprehensive community support and documentation, making it a safe choice for the development of the client side. Adding to this, the component-based architecture of React and unidirectional data flow make React perfect for creating responsive and maintainable user interfaces[6]. Moreover, previous experience and knowledge, along with a wish to learn more about best practices in React made it the framework of choice. It is worth mentioning that while TypeScript could have been utilized within React, it was excluded from the project purely based on experience.

**Golang**[7] often referred to as *Go* is a modern, open-source programming language developed by Google and was utilized for back-end API development during this project. Go was the language of choice for API development due to previous experience utilizing Go for REST API development, not to mention Go's robust standard library providing HTTP server and router functionality out-of-the-box. This, alongside a powerful concurrency model, made endpoint creation and request handling straightforward from the get-go, making it perfect for API development.

**Python**[8] was selected for implementing the Fuzzy Set Theory algorithm due to its extensive support for mathematical computing. Python's readability and concise syntax are beneficial for developing complex algorithms and the comprehensive support of libraries, such as NumPy for numerical computing facilitate efficient algorithm implementation. Python's extensive community usage and support also make it a robust language that probably will have continuous support for a while. Additionally, the support of easy integration of API endpoints makes it easy to connect to the rest of the system, despite the use of Golang for the main API.

In summary, the selection of React, Golang, and Python was driven by their respective strengths in handling specific aspects of the development process - React for its power as a single-page application development framework, comprehensive community support and documentation, Golang for efficient server-side logic, and Python for algorithmic processing. The tools and technologies are carefully chosen to ensure that the project leverages the best available resources to meet all the technical requirements in all parts of the system and achieve its overall objectives efficiently.

In addition to these programming language choices, a couple of tools were chosen for design and modeling, respectively *Figma* and *LucidChart*.

**Figma**[9] was selected for implementation of the low-fidelity prototype. Figma

[4] `https://react.dev/`
[5] `https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/`
[6] `https://www.geeksforgeeks.org/what-are-the-advantages-of-react-js/`
[7] `https://go.dev/`
[8] `https://www.python.org/`
[9] `https://www.figma.com/about/`

is an effective web design tool that makes the creation of interactive user interface prototypes easily possible. Previous experience with utilizing Figma for such cases made it the natural design tool of choice.

**LucidChart**[10] was utilized throughout all the three phases. LucidChart allowed for quickly and seamlessly visualize complex interactions and relationships within the system, and between the user and system. In this way, planning and understanding the activity flows before beginning development, while also keeping track of an overarching view of what had been developed, was possible.

Utilizing the hybrid methodology (Figure 4.1) allowed for rapid adjustments and scrapping of ideas underway while maintaining the overarching goal, making meeting the product requirements more achievable with smaller steps and adjustments along the way. While not having strict sprints planned, breaking the problem down into features which in turn were represented by their own branch allowed for smooth development. The weekly supervision meetings served as an informal sprint stand-up meeting and retrospective throughout the development period, where any issues, progress, and future directions were discussed. Their role was more towards being a stakeholder rather than a developer, resulting in the meetings mainly being on an overarching level focusing on tool needs. This approach helped significantly in achieving the goals and requirements that were defined for the tool.

For version control, GitHub[11] was utilized, enabling running Git[12] in the cloud. This allowed for version control of the tool, and branch control allowing for working on different functionalities simultaneously, serving especially useful in situations where issues were encountered in one feature. The branches were as previously stated mainly divided into feature branches, making the *main* branch serve as the stable version. Upon feature completion, the branch was merged into *main*, ensuring no conflicts, and thereafter deleted.

For programming, various IDEs were utilized which significantly improved development control and speed. This allowed easy integration of version control, code navigation, and code completion suggestions, enhancing development speed.

## 4.3 Design Phase

Throughout the design phase, the focus was on analyzing the requirements and developing a low-fidelity design based on these requirements. This phase is deemed particularly important because the design acts as the interface between the problem space and solution space [66]. During this phase, sketches of the user interfaces and the process flow of the tool were created. These were in turn made into low-fidelity user interface designs utilizing Figma. In parallel, the planned process flow of the tool was drawn out in LucidChart to ensure that all aspects of

[10] `https://www.lucidchart.com/pages/`
[11] `https://github.com/about`
[12] `https://git-scm.com`

the decision-making process were covered. Enhancements to these three activities were made upon intermediate feedback sessions with the framework developers, however, the larger refinements happened after the feedback session with the practitioner (Figure 4.3).

### 4.3.1 Sketches and Low-fidelity User Interface

In the development of the decision-support tool, the design phase focused on rapid iterations and user involvement from the beginning. The process began with simple sketches to capture initial ideas, quickly transitioning to low-fidelity user interfaces (further referred to as *low-fidelity design*) created using Figma. This approach allowed the exploration of concepts and gather user feedback without the time and resource investment required for high-fidelity design or tool development. An example of the process from sketch to a low-fidelity design in Figma is shown in Figure 4.4.

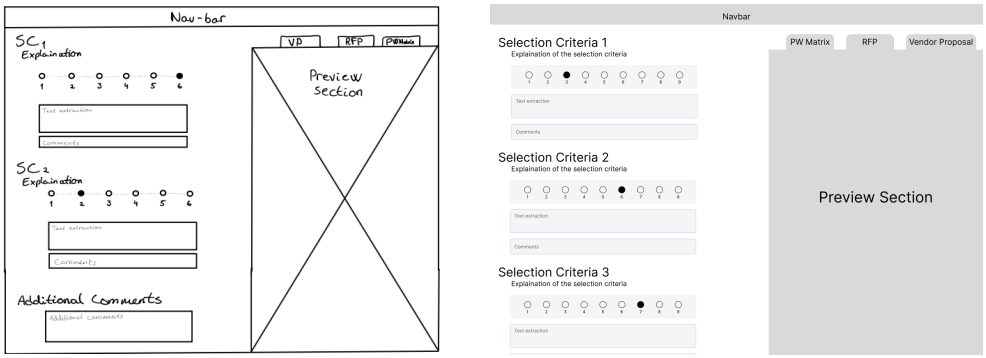


**Figure 4.4:** From Sketch to Low-fidelity Design

By utilizing this strategy the development time was significantly reduced. By addressing potential usability and functional issues during these initial stages, it minimized time-costly revisions during the coding phase, streamlining the entire development process.

In the low-fidelity designs, certain features were prioritized based on their impact on the user experience and the core functionality that was envisioned for in the tool and its requirements. These features were chosen through a combination of feedback from the framework developers and practitioner, along with strategic design decisions aimed at enhancing the tool's utility and user engagement based on literature findings which is discussed in Section 2.3.

The inclusion of these prioritized features in the designs, assisted by various UML diagrams (Section 4.3.2), allowed the refinement of the tool's interface and workflow, ensuring that the final product was both intuitive and effective. This approach not only clarified the tool's functionality for users but also provided the developer with a clear blueprint for subsequent phases, in all probability saving significant amounts of time.

### 4.3.2 UML Diagrams

As briefly addressed, a part of the design phase included the generation of various diagrams, including *use case diagrams*, *activity diagrams*, *sequence diagrams*, and an initial *architecture design* of the system (Architecture design is further detailed in Section 4.4.1). The generation of these diagrams provided multiple benefits to the project:

**Clarification of User Processes**: Activity diagrams helped in visualizing the flow of operations and the sequence of steps involved in different processes, such as for instance which steps should happen during the decision-making process. This visual representation clarified complex processes, ensuring that a shared understanding of how the user should be able to act throughout the system was shared among the developer and practitioners both before, during and after the implementation phase.
**Identification of Potential Issues**: Sequence diagrams were key in identifying and addressing potential integration and sequence issues between different system components at an early stage.
**Enhancing System Clarity**: Sequence diagrams further eased the process of having a visual illustration of the connections and interactions between the various system components during more complex system processes. This served particularly useful in cases where all the system components operated within the same process and in identifying which processes happened when and where.
**Overarching Architecture Design**: Clearly defining the overarching system architecture with its respective components, considering the chosen architecture design patterns helped planning out the system and which parts should be connected to what.
**Documentation and Future Reference**: Finally, these diagrams provide a valuable documentation resource that can be referred to throughout the lifecycle of the system. They serve as a detailed reference point for future development phases, maintenance, or even for training new team members if the project is to be scaled.

Utilizing the design phase effectively to map out user journeys, user interface designs, and architectural processes is crucial for the success of a system that accurately addresses the users' problem space. This phase serves as the interface between identifying problems and formulating solutions. Similarly, just as an API is essential for effectively mapping data and user interactions, thorough design considerations and a deep understanding derived from them are essential. Without these, the connection between the problem and the solution is more likely to fail. This again underscores the role of design as a bridge between the problem space and the solution space.

An example of how a user-journey was planned for and modeled in the tool is illustrated in the use case diagram (Figure 4.5) along with the corresponding activity diagrams (Figures 4.6, 4.7, 4.8) for each use case. Furthermore, an ex-

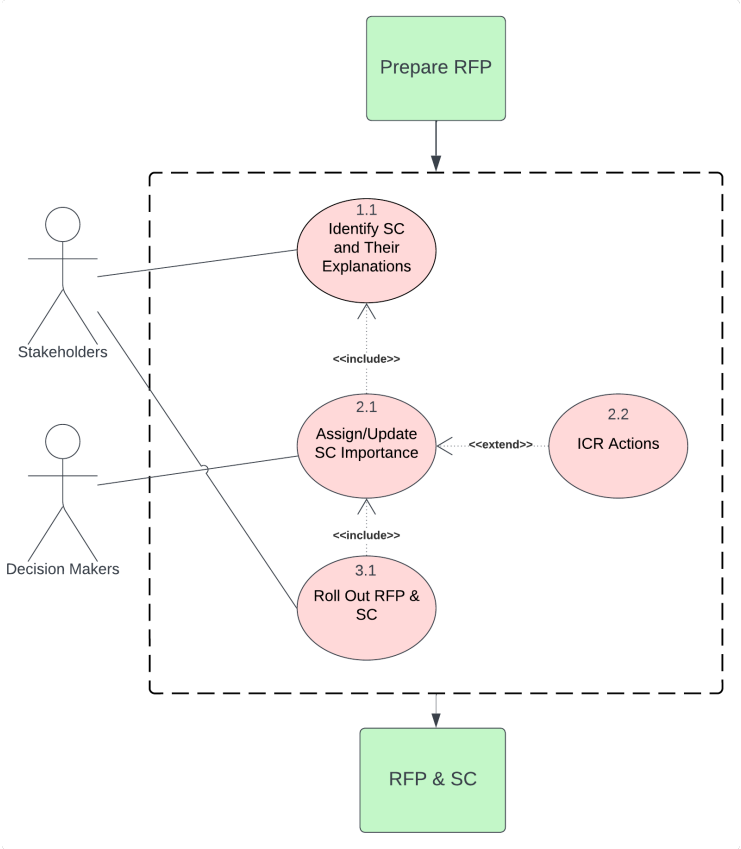


**Figure 4.5:** Identifying, Assigning, Assessing and Rolling Out SC

ample of how sequence diagrams were defined for the ICR handling can be seen in Figures 4.9 and 4.10. In the proposed example, the user journey is defined from

> *the RFP is prepared and a project is set up in the system*

until

> *the RFP and selection criteria (SC) are ready to be published*.

## 4.4 Implementation Phase

The final phase in the hybrid development method followed was *implement*. This phase served as the concluding development step for this thesis, linking the problem and solution spaces through the creation of the actual product aimed at addressing the identified issues. In this section, the system's architectural design, user interface, and tool process flow will be outlined. Further, the aspects related to separation of duties and access control will be discussed and concluded with details on the implementation of the Fuzzy algorithm.

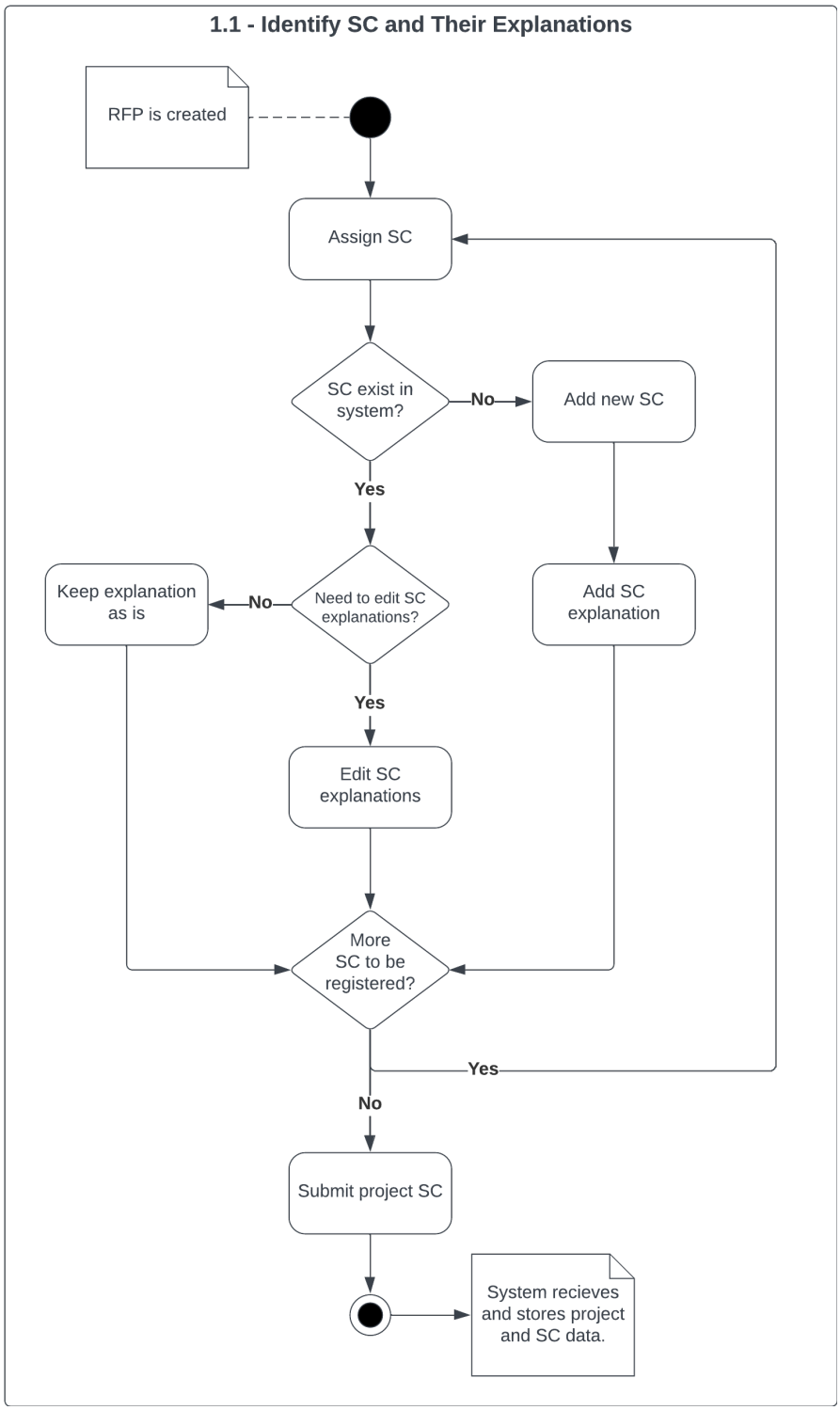


**Figure 4.6:** Activity - *Identify SC and Their Explanations*

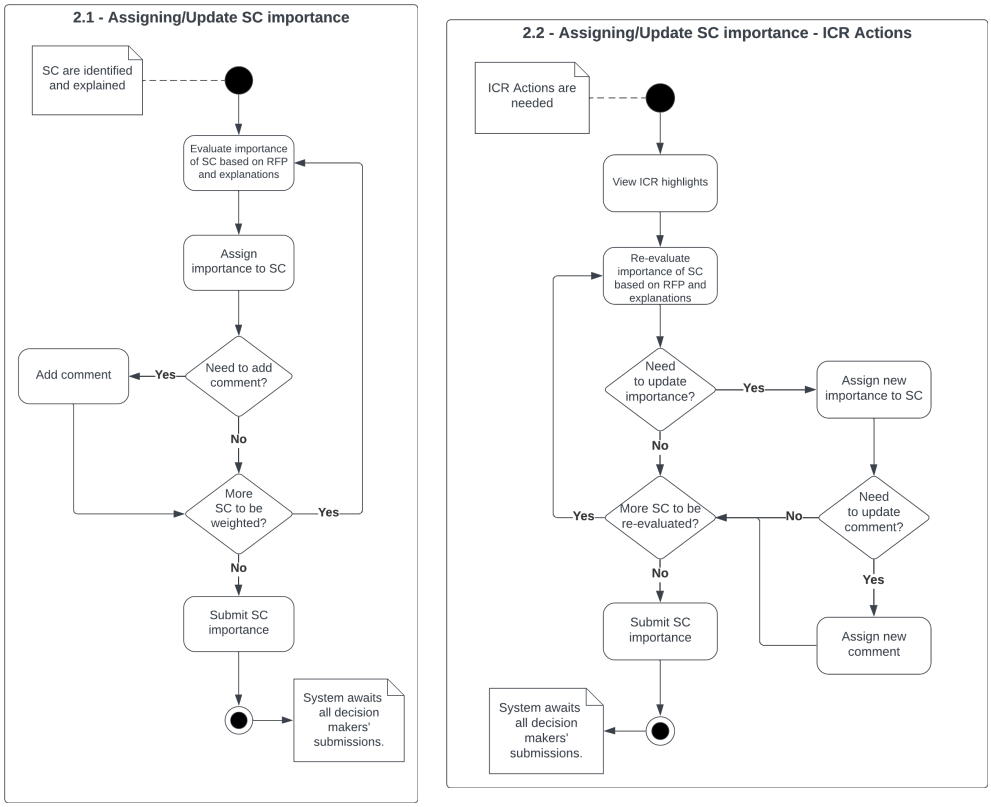


**Figure 4.7:** Activity - *Assign/Update SC Importance (w/ or w/o ICR Actions)*

### 4.4.1 Architecture Design

The architectural design of the application considers four elements as highlighted in Figure 4.11, namely; *Client*, *API*, *Storage*, and *Algorithms*.

The user interacts with the client based on their authorization status, as discussed in Section 4.4.3. The client is the web application responsible for all interactions between the user and the system, streamlining the decision-making process, and presenting the data, therefore it serves as the *presentation layer* (Figure 4.12). The API serves as the engine under the hood of the application, orchestrating on the server side. Access to this is restricted to a personal access token and is the only part of the server-side exposed to external connections, more specifically only the client. The API serves as the middle layer where business logic is processed, making it belong to the *business layer*. This includes handling requests from the client, interacting with the storage layer to retrieve or update data, and executing business rules. The API is accessed through a User Authentication Handler which makes the layer also manage authentication and authorization, acting as a gateway for client requests. Extending the business logic-layer the ICR and Fuzzy algorithms are connected to the API through their own, internal endpoints and are used to process data and decisions within the system. Lastly, the stor-

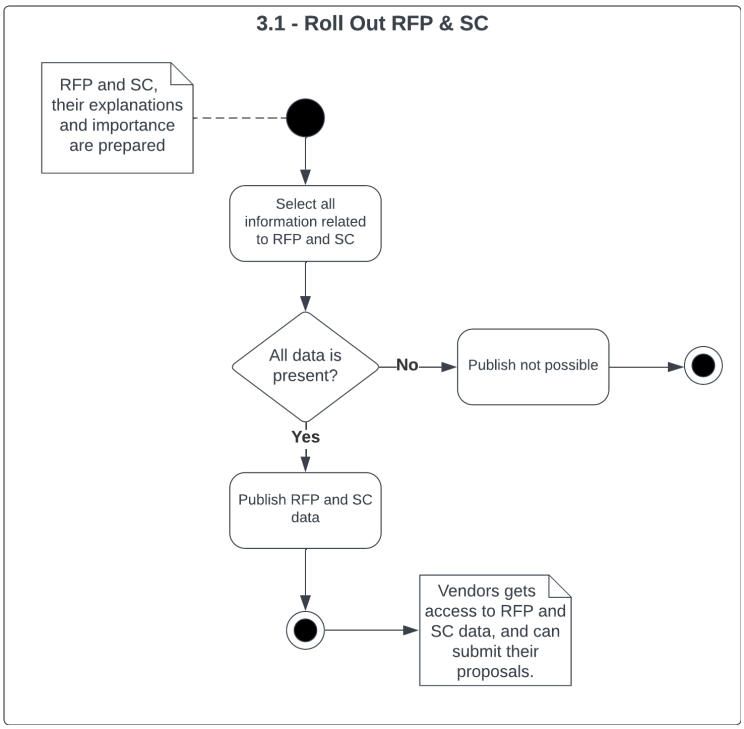


**Figure 4.8:** Activity - *Roll Out RFP & SC*

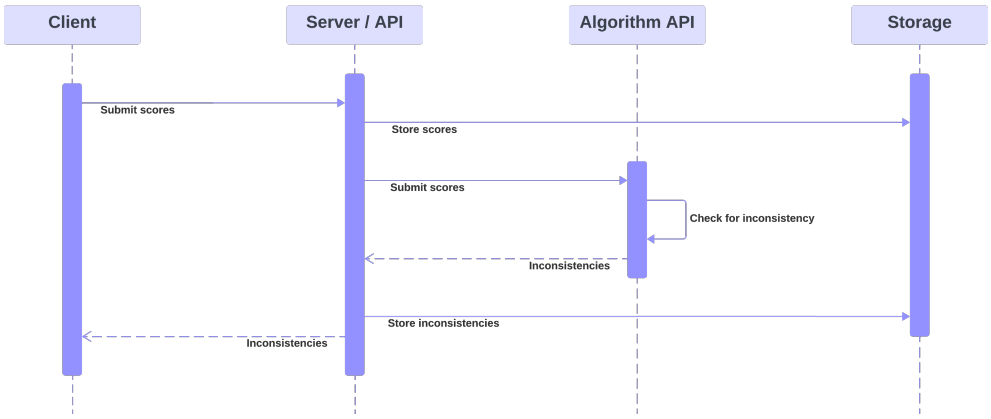


**Figure 4.9:** Sequence Diagram of System ICR Inconsistency Handling

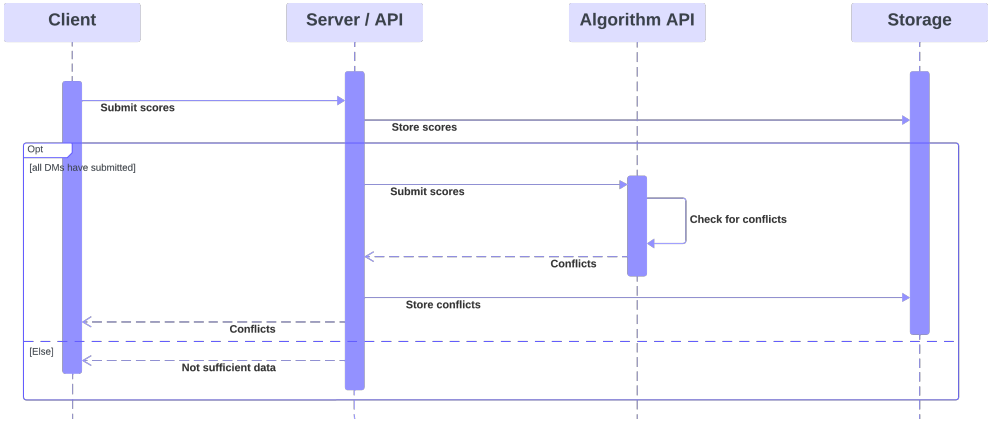


**Figure 4.10:** Sequence Diagram of System ICR Conflict Handling

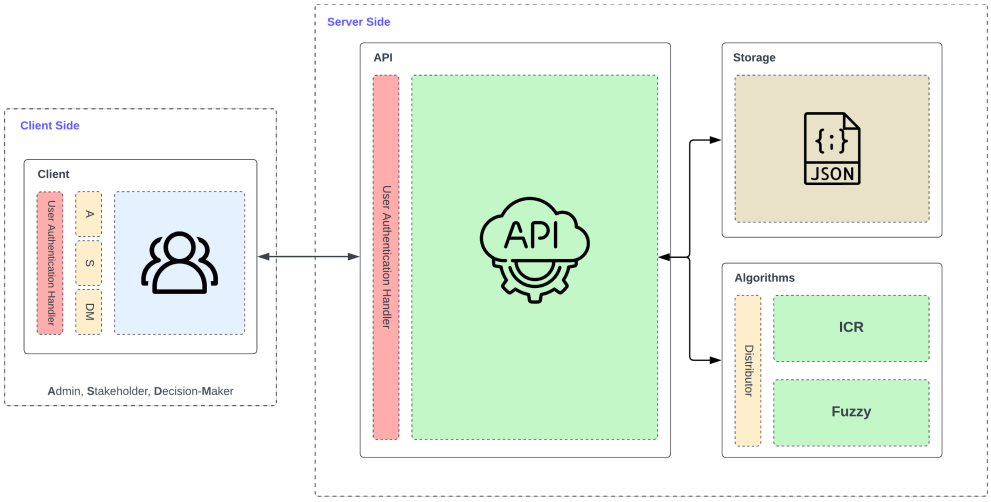


**Figure 4.11:** Decision-Support Tool Architecture Design

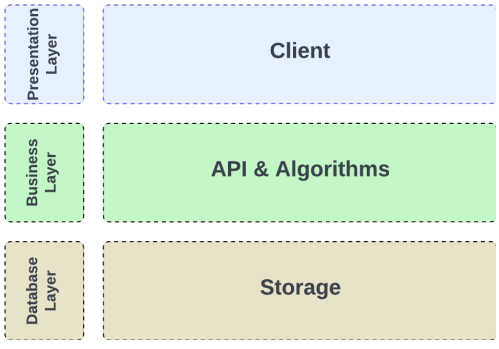


**Figure 4.12:** Layered Architecture Separation

**Table 4.3:** Requirement Address Verification (1/5)

| **Req #** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Addressed** | ✓ | | | | | | | | ✓ | | | | | |
| **Req #** | 15 | 16 | 17 | 18 | 19 | | | | | | | | | |
| **Addressed** | | | ✓ | | | | | | | | | | | |

age component belongs to the *database layer*, storing the data persistently on the server side. In the current version, the data storage happens by reading and writing to *json* files. The layered separation has multiple benefits of the application, such as separation of concerns, increased maintainability, scalability of separate layers without impacting other layers, and flexibility in technology choice where various layers can have the language best suited for their use. Furthermore, the system is designed with extensibility in mind, allowing for easy extension or removal of functionality. In this way the non-functional requirements defined for the tool (Section 4.2.1) - *usability*, *security*, *extensibility*, *maintainability*, and *scalability* are prioritized.

### 4.4.2 User Interface Design

As outlined in the Requirements Section (See Section 4.2.1), the primary objective of this tool was: *the tool should enhance and streamline the decision-making process and associated functionalities for practitioners by improving usability*, and therefore guided the entire development process. To assist in getting a comprehensive understanding of the tool's functionality and its evaluation by practitioners, as elaborated on in Section 5, a detailed walk-through of the user journey throughout the tool is essential. The steps are in turn linked to the requirements as detailed in Section 4.2.1.

The system currently supports two primary user roles: the *stakeholder* and the *decision-makers*. Each role encompasses distinct responsibilities and tasks which was discussed in Section 2.1). The stakeholder in this tool are responsible for project setup and the final vendor selection, while the decision-makers serve as the evaluators responsible for criteria and vendor capability evaluation.

#### Project Initiation and Setup

*The keystone company has some new needs which they have to solve. They have agreed to outsource the task to another vendor and have created the project's RFP.*

Upon logging in, stakeholders are directed to their dashboard. From here, they can navigate to the project setup where they can initiate a new project. This step involves entering essential project details such as the project name, and assigning stakeholders and decision-makers, as depicted in Figure 4.13.

Following the initial setup, the project is displayed on the stakeholder's dashboard, highlighting the sequential steps required. The first task for the stakeholder

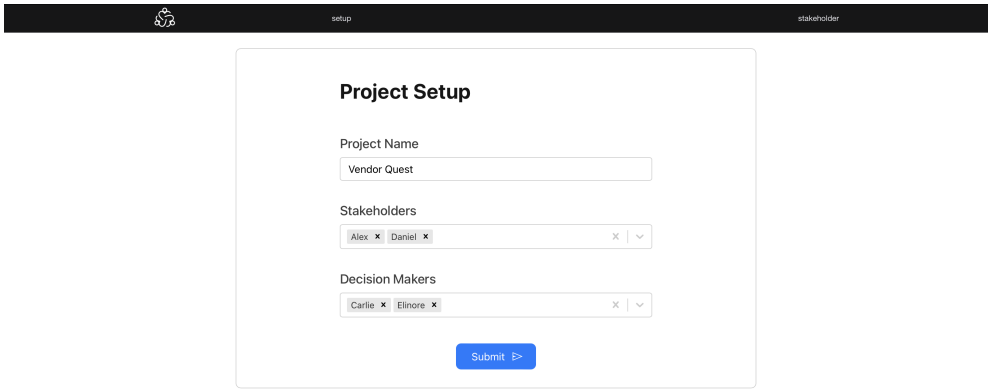


**Figure 4.13:** Project Setup Screen

is to upload the project's RFP directly from their file system, progressing to the definition of selection criteria (Figure 4.14). This process allows stakeholders to select existing criteria provided by the system through a drop-down or to add new ones. Each selected criterion's default explanation is displayed, which stakeholders can modify to align with the project's specific needs. Once finalized, these criteria and their explanations are submitted for decision-maker importance evaluation. The requirements satisfied at through this implementation is highlighted in Table 4.3

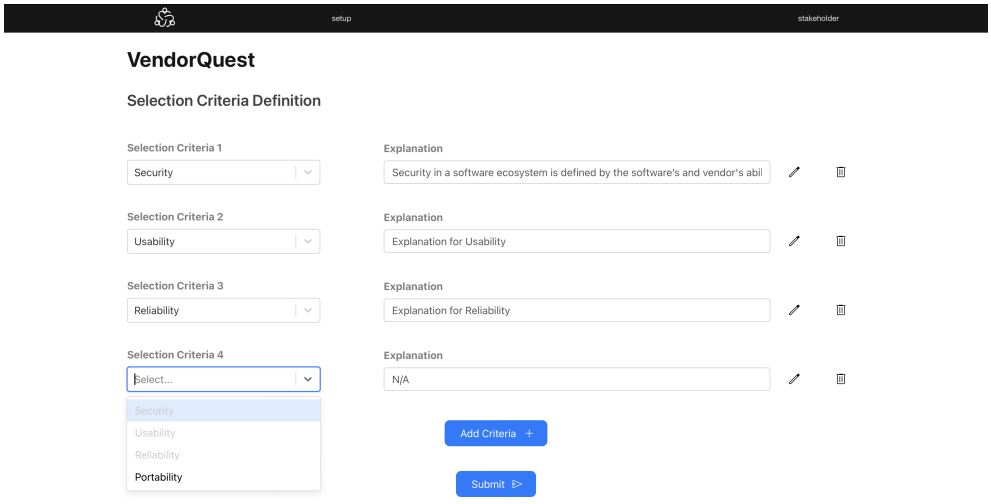


**Figure 4.14:** Criteria Definition Screen

**Table 4.4:** Requirement Address Verification (2/5)

| **Req #** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Addressed** | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | |
| **Req #** | 15 | 16 | 17 | 18 | 19 | | | | | | | | | |
| **Addressed** | | | ✓ | | | | | | | | | | | |

### Criteria Importance Assessment

*Selection criteria are assigned to the project and requires an assessment of their importance.*

Before the RFP and selection criteria are distributed to potential vendors, their importance must be evaluated by decision-makers. This task is facilitated through a dedicated evaluation page accessible via their dashboard, where the criteria are presented for assessment (Figure 4.15). The evaluation interface is designed to minimize the cognitive load by separating criteria-related data and the RFP into distinct sections of the screen, enhancing the accessibility and usability of the information. Gathering all the information within one interface without overwhelming the decision-makers was and should be of great priority as the literature also emphasizes.

Notable changes have been made on the interface side from the traditional pairwise comparison method used in the AHP, moving the heavy logic to the system and providing the decision-maker with a more isolated textual scoring system. This modification supported by both practitioner feedback and literature, helps reduce complexity and mental load on the decision-maker. The interface systematically guides decision-makers to focus on one base criterion at a time, comparing it against all others before proceeding to the next. This sequential evaluation ensures thoroughness and that the decision-makers maintain focus. Both linguistic and numeric scoring options are available to accommodate diverse user preferences, with a particular emphasis on linguistic values to enhance usability and intuitiveness. Decision-makers also have the option to invert scores and add comments to document their rationale, improving transparency and aiding in conflict resolution. Furthermore, consistently displaying the criteria explanations ensures that all decision-makers possess a uniform understanding of the criteria, further simplifying the decision-making process and making it more accessible.

Upon completion of the criteria importance evaluations, the system automatically identifies and highlights any inconsistencies or conflicts. These must be addressed and resolved before finalizing decisions. The dashboard is dynamically updated to reflect the resolution status of these issues (Figure 4.16). The issues are further highlighted within the evaluation screen, as illustrated in Figure 4.17. This process is iterative and continues until all issues are satisfactorily resolved. The requirements that is satisfied in this step is highlighted in Table 4.4.

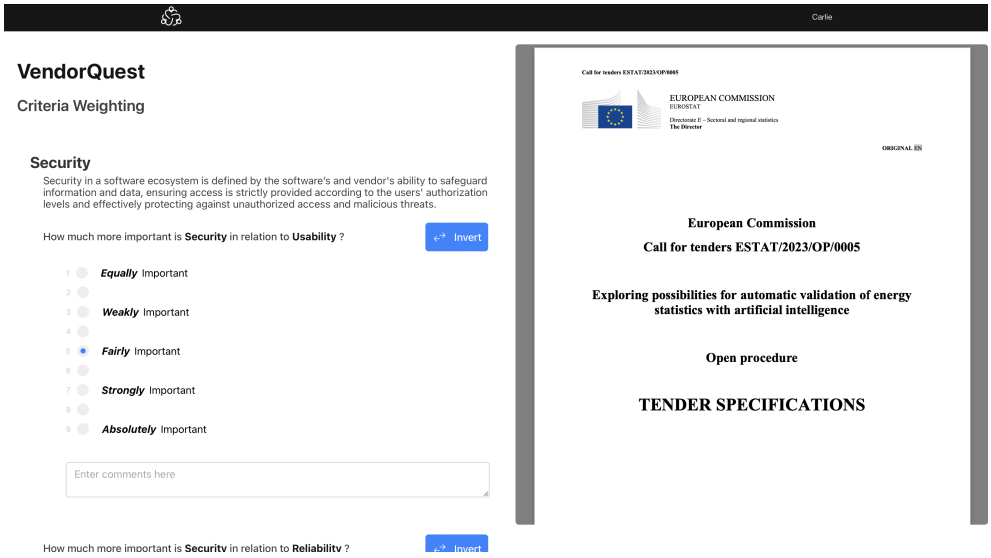


**Figure 4.15:** Criteria Evaluation Screen

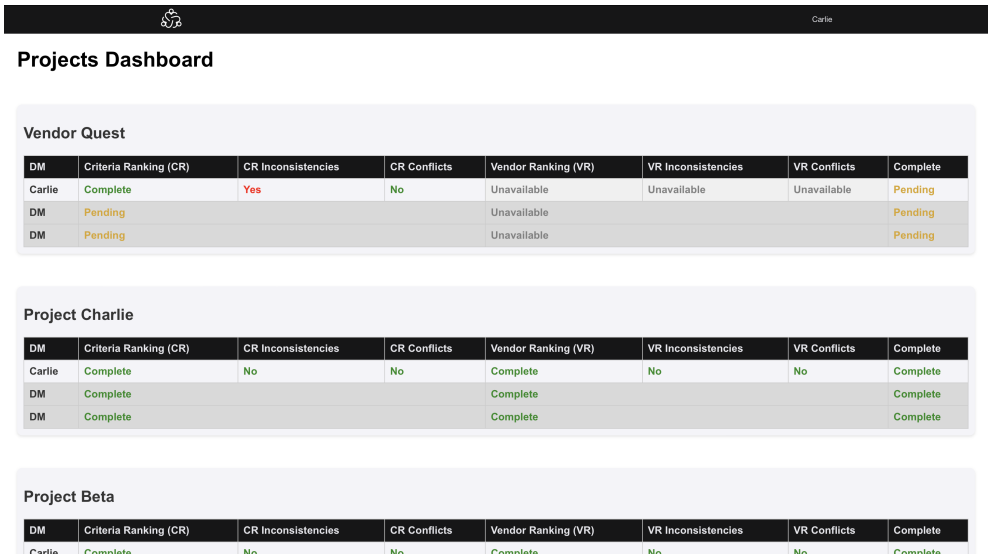


**Figure 4.16:** Dashboard With Inconsistency Screen

### Vendor Capability Evaluation and Selection

*Upon project setup, RFPs, and selection criteria are rolled out. Vendors respond with their information and proposals to the outsourcing task.*

Following the evaluation of selection criteria, vendors submit their proposals, which are then assessed by decision-makers. Similar to the criteria evaluation, vendor capabilities are scored against each criterion within the same user interface, which now includes a text extraction feature for better document manage-

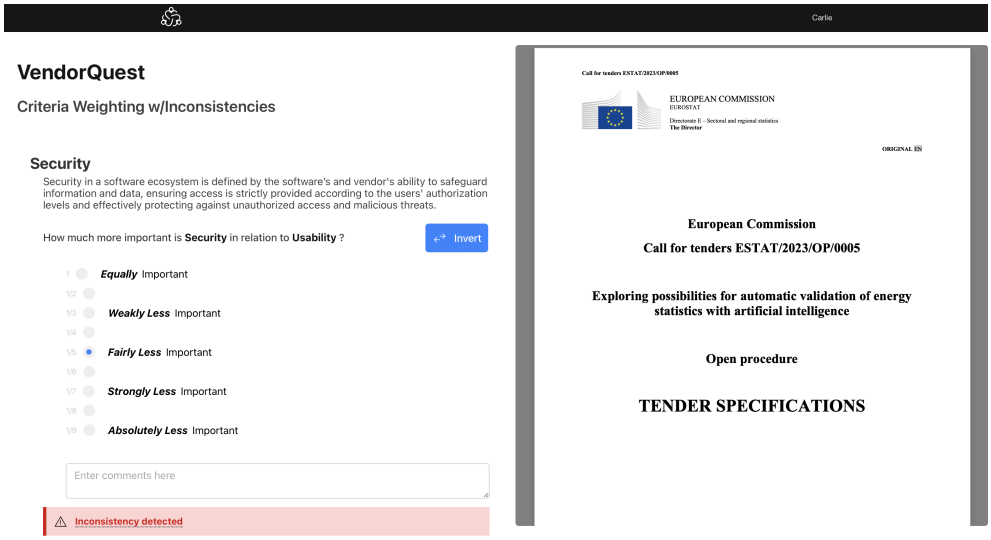


**Figure 4.17:** Criteria Evaluation with Inconsistency Highlight

ment.

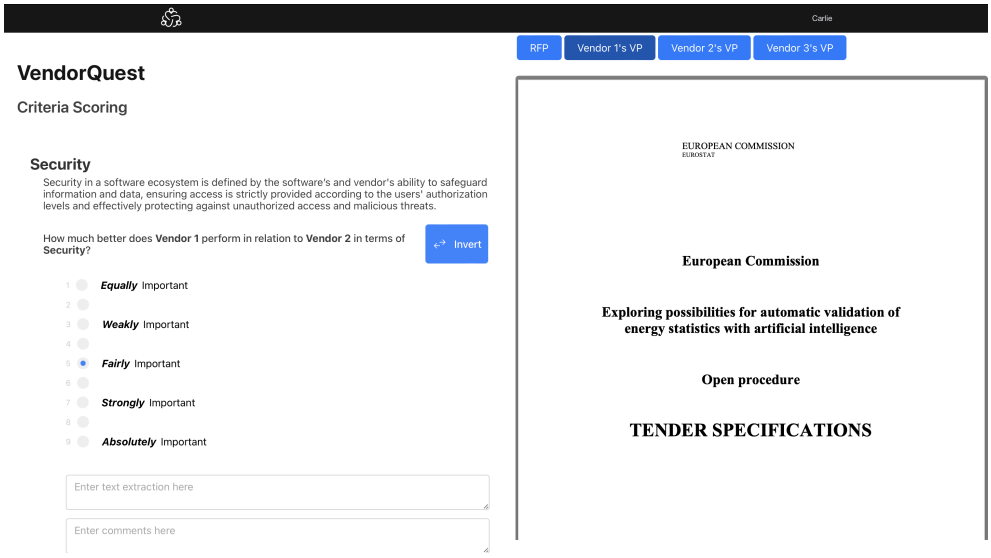


**Figure 4.18:** Vendor Capability Evaluation Screen

Upon completion of the vendor-capability evaluations, once again, any inconsistencies or conflicts are highlighted in the same way as in Figure 4.16 and 4.17, and must be resolved before finalizing the decision. In addition to the requirements already satisfied through other steps which also is satisfied here, two additional requirements are accommodated through this implementation as seen in Table 4.5.

**Table 4.5:** Requirement Address Verification (3/5)

| **Req #** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Addressed** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| **Req #** | 15 | 16 | 17 | 18 | 19 | | | | | | | | | |
| **Addressed** | | | ✓ | | | | | | | | | | | |

**Table 4.6:** Requirement Address Verification (4/5)

| **Req #** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Addressed** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| **Req #** | 15 | 16 | 17 | 18 | 19 | | | | | | | | | |
| **Addressed** | | | ✓ | | | | | | | | | | | |

**Final Vendor Selection**

*All decision-makers have completed their evaluation task.*

Finally, using the fuzzy calculation algorithm, the system ranks the vendors based on the consolidated scores from all decision-makers. The final ranking is presented to the stakeholders, who are tasked with making the concluding vendor selection (Figure 4.19). To aid in this critical decision-making process, detailed visualizations such as graphs are provided to summarize and effectively represent the rationale behind the ranking to the stakeholders. These visual tools not only ensure transparency but also help stakeholders to quickly grasp complex data, identify trends, and detect any anomalies or inconsistencies in the scoring. Once the stakeholders have reviewed the data and reached a common consensus, they select the preferred vendor and initiate contact to finalize the decision-making process. This step marks the completion of the vendor selection phase, ensuring a deliberate and well-informed choice. Two more requirements are accommodated through this implementation as shown in Table 4.7.

### 4.4.3 Separation of Duties and Access Control

One of the important features in the system is the separation of duties and access control handling. There are two considerations done regarding this, namely; giving the clients access to the web-app based on their user-data and secondly, that only authorized users should be allowed to access the back-end of the application. This was one of the features necessary as feedback from the practitioner during the user-interface evaluation and the literature clearly stated the differences in which roles should be allowed to perform which tasks.

Figure 4.20 details the sequence of logging in to the application. This sequence shows how the system handles credentials and how the API provides the user with a Cookie token which are then used for all the subsequent request issued from the client to the API.

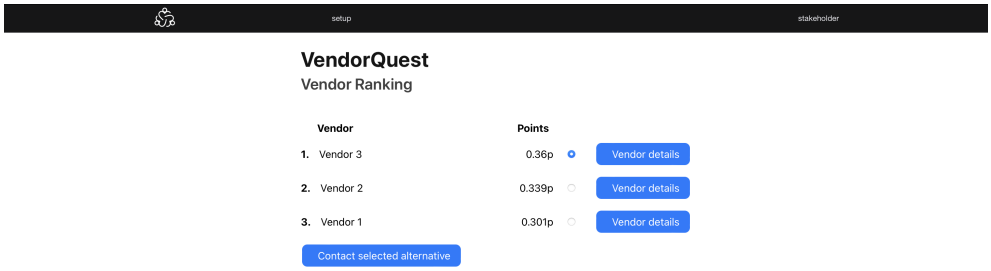


**Figure 4.19:** Final Vendor Ranking Screen

**Table 4.7:** Requirement Address Verification (5/5)

| **Req #** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Addressed** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| **Req #** | 15 | 16 | 17 | 18 | 19 | | | | | | | | | |
| **Addressed** | | ✓ | ✓ | | | | | | | | | | | |

### API

For the API and access to the back-end, three sets of middleware were set up. First, for internal accesses only, one middleware was setup on the API, only allowing request from the algorithm endpoints. Second, one middleware open to everyone, which allowed for accessing the user login and registering new users, only limiting on the domain in which the request comes from. Lastly, for comprehensive access to the entire API, a middleware both considering domain in which the request came from and whether the user who raised the request is authenticated with an authenticate token, utilizing JSON Web Tokens (JWT)[13]. This allowed for user authorization upon requests. While this adds a layer of security, the way it works now in the proof of concept is that if such a token is present in the request, full access is granted to the API which is not good in a production setting. While considering it in this setting is outside of the thesis, one thing that should be done as a minimum before further development is access control to endpoints based on user roles as well, not only authorized user or not.

[13] `https://jwt.io/introduction`

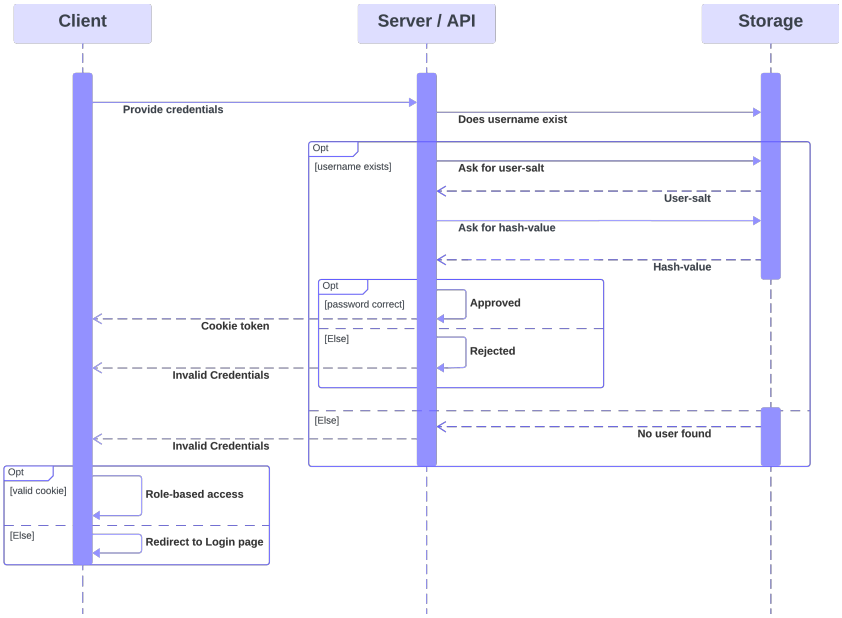


**Figure 4.20:** User Login Sequence

### Client

For client side access, the utilization of AuthProviders and ProtectedRoute was utilized. As illustrated in the architecture design (Figure 4.11), the client separates duties/access based on user roles. The AuthProvider serves as a context provider for authentication-related data. It utilizes React Context API to provide a global state that can be accessed by other components without prop-drilling[14]. By using such an approach the user's authentication token is stored and utilized throughout the application both for access control client-side and for requests to the API. Some information is also possible to retrieve from the token such as user role and name. Additionally, the token carries an expiry claim checking whether the current token is valid and not expired. The ProtectedRoute acts as a wrapper for the routes that require user authentication and authorization. It uses the context provided by AuthProvider to determine if the user should be allowed to access certain routes based on the user's role. If the user is not authorized the ProtectedRoute handles the redirection of the user. Overall, AuthProvider provides essential authentication functions and data to the components, while ProtectedRoute ensures that certain parts of the application are accessible only to authenticated and authorized users.

## 4.4.4 Fuzzy Set Theory

As the last part of the implementation, discussing the implementation of the scaling of the framework to utilize Fuzzy is required. As highlighted in Section 2.2.3,

[14] `https://www.geeksforgeeks.org/what-is-prop-drilling-and-how-to-avoid-it/`

the integration of Fuzzy Set Theory into the ICR framework is based on Ayhan's research [56], which explores the application of FAHP in supplier selection. While Ayhan outlined the necessary steps for employing FAHP, no actual coding implementation was provided. Consequently, this section is dedicated to the implementation of Fuzzy AHP within the context of this thesis using Python, as introduced in Section 4.2.3.

If there is an interest to see the details of the implementation, visit the codebase on GitHub[15]. On a top-level, excluding some of the intermediate steps in weight calculation which is detailed in Ayhan's work; it begins with processing the input matrices for criteria scoring (criteria-criteria) and vendor capabilities, which are provided in *csv* format for all necessary comparisons. This includes a criteria matrix and all vendor-capability matrices. It's important to note that the process is divided into two distinct phases: criteria weighting followed by vendor capability scoring, each utilizing similar yet specific data inputs and outputs. Initially, the input scores are converted from the Saaty scale to fuzzy numbers for all entries. This conversion is necessitated by the API's initial design, which currently stores decision-makers' scores as Saaty values due to the required format of the ICR algorithm. This architectural decision mandates that the mapping to fuzzy numbers occur at this level of the implementation using the *Saaty-Fuzzy Triangular Scale* mapping as presented in Section 2.2.3, Figure 2.6. Subsequent steps involve calculating fuzzy weights via geometric means, which are then defuzzified and normalized. This is first done for the criteria-weighting process, then repeated with the vendor-capability matrices for each criteria. The final ranking is made up by multiplying the criteria weights with the respective vendor scores for that criteria, summing and lastly normalizing these scores. The resulting normalized values determine the vendor scores, identifying the vendor with the highest score as the most suitable choice. Figure 4.21 and Figure 4.22 details the steps involved.

[15] `https://github.com/MACS490-TMM/Tool`

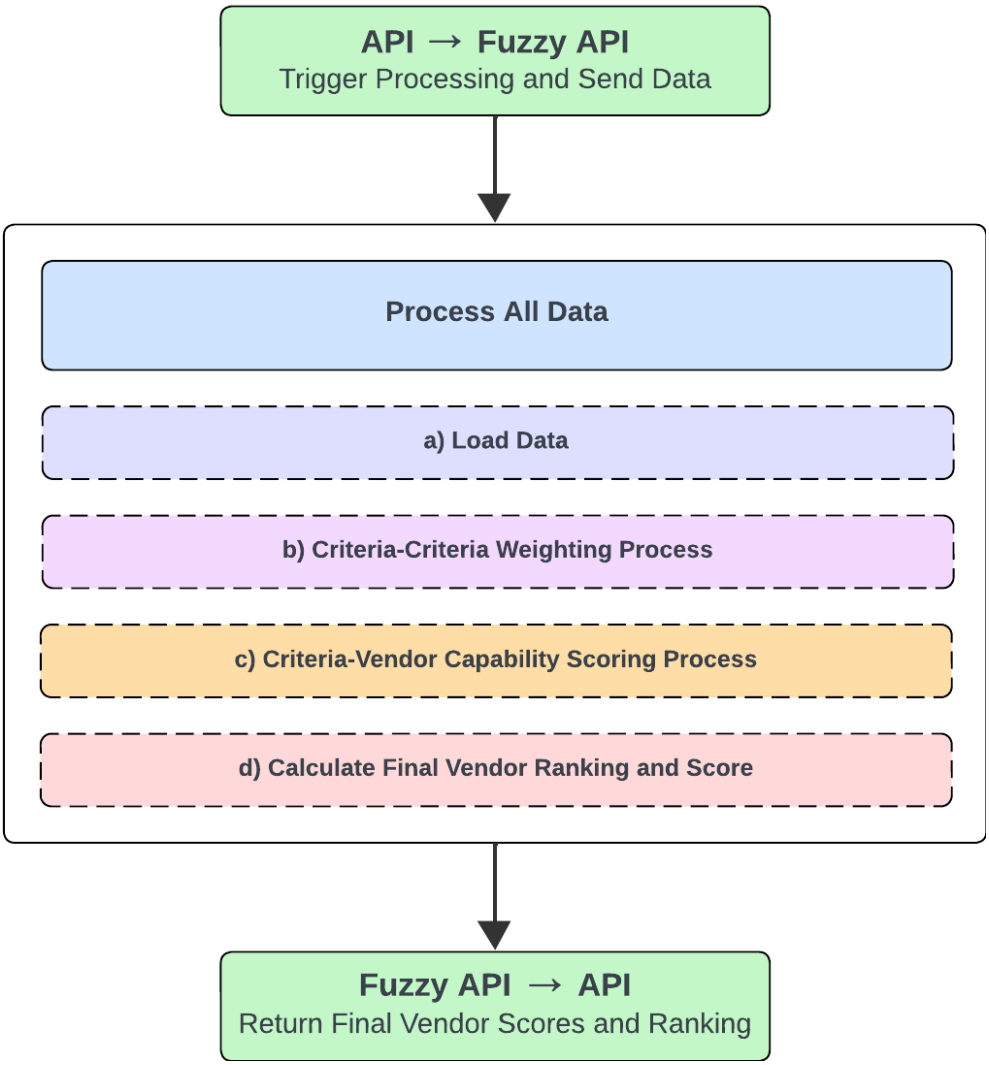


**Figure 4.21:** Top Level Fuzzy Implementation

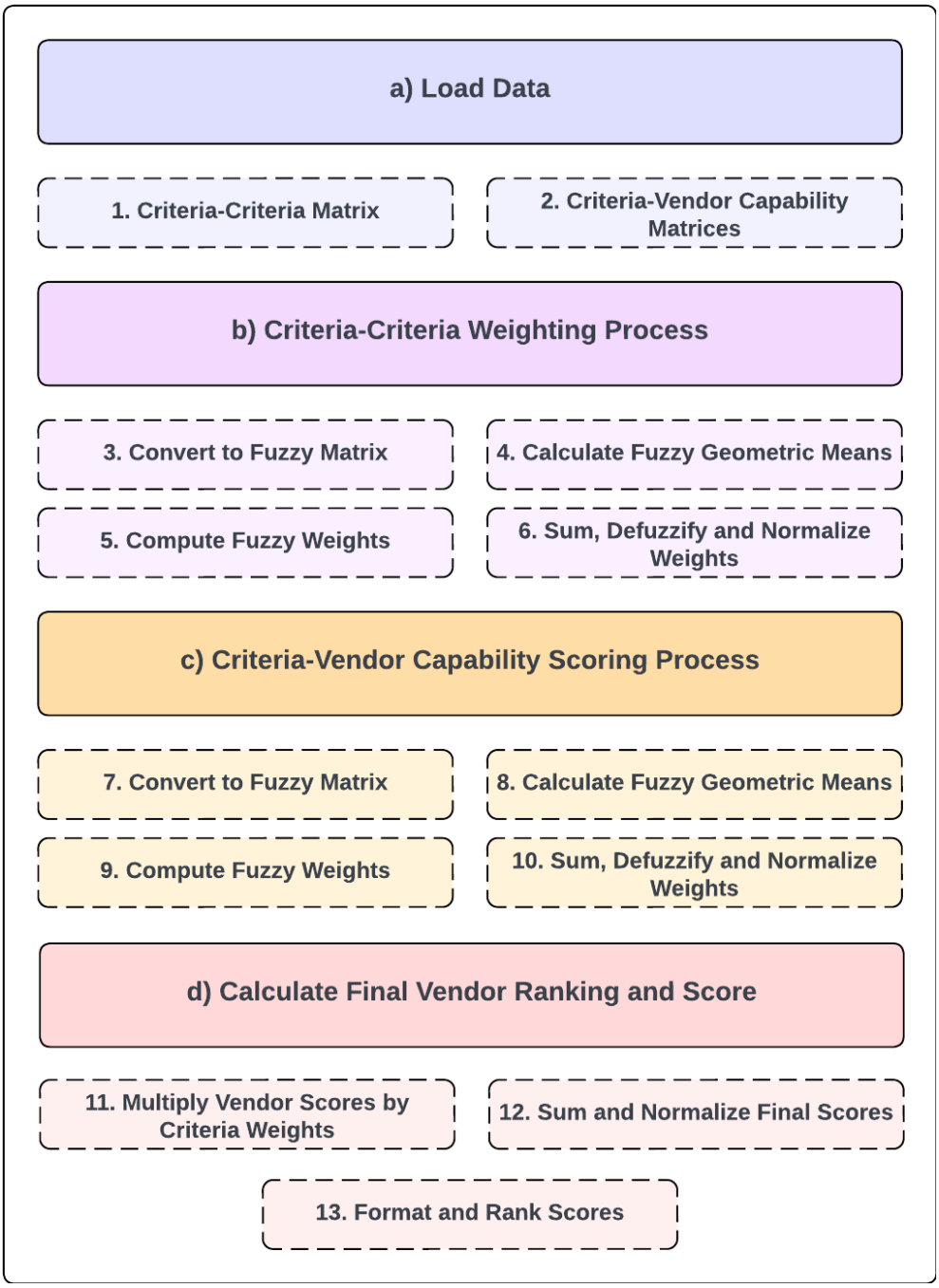


**Figure 4.22:** Detailed Fuzzy Implementation Based on Figure 4.21

# Chapter 5

# Tool Evaluation

The evaluation of the tool are split into two distinct parts: low-fidelity design evaluation, followed by the final tool evaluation (Figure 5.1). Consequently, this chapter outlines the practitioner evaluation across two phases. The first phase focuses on the design and process flow of the low-fidelity user interface, whereas the second phase addresses the development and assessment of the final tool.

## 5.1 First Phase Evaluation

In this section, the focus is on the evaluation conducted with an expert practitioner evaluating the initial low-fidelity user interface design and proposed process flow. Initially, the selected evaluation strategy and the background of the practitioner will be presented, before a summary of the main findings of the evaluation are presented and summarized.

### 5.1.1 Evaluation Strategy

As introduced in Section 3.4, the evaluation strategy of choice was a semi-structured interview. This strategy would offer the opportunity to gather detailed feedback and understand the context in which the tool is used. The semi-structured nature would also allow the exploration of unique or unexpected directions that would be especially useful at this early stage.

An interview guide was prepared to have a structured baseline of what to consider during the interview. This guide included questions regarding their vendor selection processes, satisfaction with the current design and tool processes (visualized with activity diagrams and use-case diagrams), and preferences for such tools. The interview guide can be seen in Appendix C.1. The interview was conducted using MS Teams and lasted for approximately one and a half hours. The details about the practitioner are presented in Table 5.1. The outcome of the interview was an evaluation of the proposed tool process flow and the user interface design, including feedback on what should be excluded and what should be included in the upcoming phase to make it more suitable for their practices.


**Figure 5.1:** Action Research Methodology, Evaluation Phase

**Table 5.1:** Details of Interviewed Practitioner for First Phase

| Participant | Professional Responsibilities | Organization Type | Decision-making Experience | Location |
|---|---|---|---|---|
| P3 | Project Manager, System Analyst, Member of IT projects Assessment Team | Ministry of Interior | 13 Years | Bahrain |

### 5.1.2 Main Findings

A summary of the practitioner's feedback is to be presented next while a more detailed look at *some* of the practitioner's feedback can be seen in Table 5.1.2.

The practitioner necessitates and validates that such a tool is required. According to the insights received, until now, decision-makers heavily rely on manual and ad-hoc processes for software vendor selection, which involve complicated tasks like writing down page numbers and manually checking each entry. This process is time-consuming and error-prone, making decision-makers wish for a streamlined process from project setup to final vendor ranking. Thus, the tool presented in this study will benefit decision-makers in automating the tasks and centralizing information to overcome these issues.

Adding to the processes being labor-intensive, the practitioner also emphasized that in practice, they lack a systematic approach, leading to inefficiencies and potential oversights in the decision-making process. Thus, one of the aids of the proposed tool will be to provide all decision-makers with a similar semi-systematic and data-driven evaluation of software vendors, in the hopes of enhancing decision-making efficiency by reducing errors. Additionally, the practitioner highlighted the significance of consolidating all decision-making data onto a single platform, emphasizing how the proposed tool can simplify the decision-making journey and contribute to its success.

Lastly, the practitioner also highlighted that prior practices did not effectively leverage visualizations in the decision-making process. Since the proposed tool carefully integrates visualizations into parts of the decision-making process, this further supports decision-makers in making more informed decisions, along with easily comparing vendor data in one place. This aspect of the proposed tool was appreciated in particular by the practitioner.

> *Two requirements that are to change for the second phase are; dropping the pair-wise matrix view as it will be unreadable and of no use when projects scale large (60-70 criteria and 20-30 vendors), and adding priority to implementing an anonymous contact solution allowing decision-makers to directly contact vendors from within the tool, whether through email or a chat function.*

These practitioner insights were brought back to the planning phase leading to refinements in the requirements and project scope. Following these adjustments, the low-fidelity user-interface designs and process flows were updated accordingly, before beginning the development of the tool.

## 5.2 Second Phase Evaluation

This section presents the details of the expert practitioners' evaluation and feedback on the proposed tool's user interface and the tool process flow. First, the selected evaluation strategy will be revisited. Second, the practitioners' background

**Table 5.2:** Insights from practitioner

| Theme | Practitioner (P3) |
|---|---|
| User interface evaluation | '...Sounds like a good idea to be honest. Very good idea. I don't know why we didn't think about an idea like this before. That's a very good idea, to be honest. I'm thinking on the possibilities of like, how can we use the same thing?' |
| | Depending on the size of the project and what kind of project it is. So 60 to 70 criteria and let's say average size of a project is having around 10 vendors. Like 60-70 criteria multiplied by ten and then comparing each vendor and each criteria with different thing it will be time consuming... |
| | '...So I'll have some questions and explanations. So the question will in the questions section, I'll know what I need to answer or select based on the question and explanation will help me in that process like exactly what I'm evaluating on that question because documentation or any criteria, differs from person to person. But if there is an explanation like what exactly are we looking for in this question, the evaluator will have to think in that aspect only.' |
| Suggested improvements | ...We do phone, emails, everything, but if the system itself is having this option in the criteria, OK I'm checking this vendor in this section and I have a question for the vendor itself, so I'll write the question and send it and the system will maybe have the e-mail of the vendor so it will go to the vendor and come back here. So everything is transparent.... |
| | ....Yeah, because for like, transparency reason also ... because we call them which is bad... I should just write an e-mail with my question, it should go to the person without knowing my name or I know his name. He just replies me back. Because in our business, we know each other, we shouldn't know who is asking what.... |
| | ...This will be helpful because we just write the page number and everything, and the person who wants to check it, they have to do it manually.... |

will be introduced. Finally, the main findings from the evaluation will be presented in detail.

### 5.2.1 Evaluation Strategy

Despite various research strategies being considered as discussed in Section 3.4, such as; conducting workshops with a set of practitioners, participant observations, or providing practitioners with a survey, semi-structured interviews were once again assessed as the best-suited evaluation strategy. This was deemed especially useful as this time more practitioners were involved which could lead to interesting digressions that either reinforced the findings of the first evaluation phase or were not addressed during that phase.

This time the interview guide initially explored the practitioners' current methods but put extra emphasis on the tool evaluation concerning usefulness and usability, as well as features and functionality. The interview guide and its details can be seen in Appendix C.2. Furthermore, two of the interviews were conducted online using MS Teams as the practitioners and researcher were not in the same location while the last interview was conducted in person. All three of which lasted for approximately an hour. The outcome of the tool evaluation was detailed feedback on usefulness and usability concerns and features and functionalities that should be in such tools, as well as additional features they would have liked to see.

In the upcoming sections, the findings from the evaluations will be presented

**Table 5.3:** Details of Practitioners Interviewed for Second Phase

| Participant | Professional Responsibilities | Organization Type | Decision-making Experience | Location |
|---|---|---|---|---|
| P1 | Decision Maker, Tender Preparation and Analysis, Director Critical System Unit | ICT Information and Communication Technologies | 15-20 Years | Turkey |
| P2 | PMO Advisor | Australian Operations for Government Contracting | 15 Years | Australia |
| P3 | Project Manager, System Analyst, Member of IT projects Assessment Team | Ministry of Interior | 13 Years | Bahrain |

addressing the feedback from three practitioners. The practitioners have been anonymized and are designated as *P1*, *P2*, and *P3*. Details about their backgrounds can be found in Table 5.3.

### 5.2.2 Practitioner Feedback

#### Current tools

As the literature had limited focus on which tools practitioners utilize in the their decision-making processes, understanding the practitioners current tools was deemed interesting. A shared consensus among the practitioners were that the decision-making process is an ad-hoc and tedious process mostly conducted through manual labor in tools like Excel or even on paper. *P3* highlighted the use of a tool to handle vendor proposal uploads from vendors, but the tool only supports in that task.

Furthermore, discussion about the existing tools, related to Excel's and the vendor proposal upload tool's strengths and weaknesses within the decision-making context arose. Limitations such as data transfer capabilities, duplication and version control is brought up as some of the key constraints of using Excel. Furthermore, the security and template of Excel is also highlighted as a limitation. One of the practitioners also hesitated in acknowledging Excel as a tool for the process, but rather just assisting the practitioners by using it (along the other documents) as the task could even be done with pen and paper and then submit the result. Additionally, the tool for vendor proposal uploads have a fee which have to be paid, making each vendor have to pay the fee which is addressed as a limitation of the tool.

The strengths related to the tools, considering Excel and the vendor proposal upload tool, are just as important to understand to comprehend the decision-

**Table 5.4:** Insights From Practitioners on Current Tools

| Practitioner (P) | Insights |
|---|---|
| P1 | We are not using any tools in the process. What we are doing is that of course we are evaluating the vendors or the subcontractors, but these are done mostly manually. I mean of course you grade in a way, but the results are only recorded in Excel or sometimes only giving better or worse on a paper. |
| P2 | A lot of the decision making process at the moment is simply done by Excel spreadsheets. |
| P3 | Actually, there is a tool that we use for the vendor to submit their proposals. We add the details of the tender, what the tender is about and what are the criteria of the submissions because there are different criteria for the different submissions. The vendors will upload their proposals then the tender is gone and before the timetable closes. For example, lets say it's the ten days window, within this window all the vendors who are interested, they were submitted. |
| | Let's say we have around 20 to 30 proposals and we need to check them. So we make Excel sheets with different scoring. |

**Table 5.5:** Insights From Practitioners on Limitations in Current Tools

| Practitioner (P) | Insights |
|---|---|
| P2 | There's a couple of challenges around data transfer and that sort of thing... and the duplication and version control and all of those stock standard issues that you have with Excel |
| P3 | And there is a fee if he has paid the fee for the submission and everything. |
| | Excel is not the most secure or have the best template to be using here to be honest. So other than that, there is no tool. I cannot consider the Excel documents as tools. They're just helping the person. You can even do it in pen and paper and submit the result. |

making context. The flexibility of Excel is brought forward as one of its strengths. Additionally, the learnability of Excel and simplicity of setup and preparation are highlighted, as most people are familiar with Excel. For the vendor proposal upload tool, the only good thing about the tool according to one of the practitioners is the assurance of data confidentiality until the tender period is over.

**Table 5.6:** Insights From Practitioners on Strengths in Current Tools

| Practitioner (P) | Insights |
|---|---|
| P2 | Excel gives you flexibility. |
| | Excel doesn't require people to understand the system, they just need a briefing on Excel which most people are familiar with... Those involved in the decision-making getting emails to log in to the systems are often uncomfortable with that process, so doing a moderated session and entering details for them in Excel is often quicker and easier, especially with more time poor senior managers. |
| P3 | The only good thing about the vendor proposal tool is currently like all the vendors, when they submitted it, they submitted online and they can be assured like no one has access to the file before the tender period is over. |

### Selection Process

Next, the selection process will be examined, specifically how essential criteria and their importance are identified, and if certain criteria are deemed generally

essential among the practitioners. The initial step involves understanding how essential criteria are determined.

Selection criteria for a project are determined in various ways based on the practitioners' feedback. Some decide on them before RFP generation, while others determine the criteria based on the RFP. The only consensus is that these influence each other, and that usually both are provided to the vendors or at least exist during evaluation and that it is a business conversation. In some cases even, only parts of the RFP and selection criteria are shared with the vendors, making up the parts this vendor should be responsible for out of a larger project scope. Lastly, the number of criteria and which criteria are included varies from project to project.

**Table 5.7:** Insights From Practitioners on Selection Process

| **Practitioner (P)** | **Insights** |
|---|---|
| P1 | They are doing just like you. They are extracting the parts and sometimes that they are doing is only sending those parts to the vendors so that they will be only responsible for that part. |
| P2 | What happens is the list gets worked into the RFP, so you may have them respond to an Excel spreadsheet or a Word document. It's going to differ and vary. |
| | Part of that is just the business conversation. You know the must, should, could or will not have... |
| P3 | From the perspective of security, if it's a security project or it's a normal project, it all depends. We don't have a fixed criteria for all the projects. |
| | It again depends. Sometimes it is 15 or 20 criteria... sometimes it goes to 100 maybe. It all depends... I have never seen a project where it's repeating itself each time. |

In presenting the decision-makers' perspectives on how the importance of criteria is determined, it becomes evident that this process is highly subjective. Practitioners note that criteria importance varies based on factors such as the decision-makers' and stakeholders' experience and expertise, the relevant business area, and the specific needs of the project. Assignments of importance range from binary yes/no decisions to more nuanced scores guided by established company guidelines. Furthermore, the significance of some criteria may be decided upon based on sub-criteria assessments. There is a general agreement among practitioners that criteria should be assigned a level of importance, which is typically determined by those experienced in the procurement process.

The final consideration of the selection process relates to whether or not there are any criteria that are considered for all projects. It appears that certain criteria are universally considered, such as the reliability and financial stability of the vendor. Other factors consistently evaluated according to one of the practitioners include the sustainability and stability of the vendors, as well as the completeness and consistency of their solutions. Additionally, according to another practitioner, security is almost always a key consideration, along with the solution's compatibility with the existing systems. The availability of training, the level of post-implementation service, and the languages used by the system and for communication are also deemed important.

**Table 5.8:** Insights From Practitioners on Determination of Criteria Importance

| Practitioner (P) | Insights |
|---|---|
| P1 | There should be some scaling. You don't have a set importance for the criteria, it depends on the nature of the project...sometimes security is important, sometimes the consistency or volume of the storage is important. It depends. |
| | ...for example,... security may have some items that naturally provide you the scaling. If you have five sub-items for security but only two items sustainability or maintainability, then it means that security will be more important. Instead of getting rates, you're categorizing the features. |
| P2 | ... the domain-experts review it from what do they need from their business area and how do they view it from an overall business need. |
| P3 | ...for example; can we integrate it into our current system?... A or B, there is nothing in between. There is no score. It's like if there is a score, I'll give it 0 or 10. |
| | ...usually we use numbers to score criteria and vendors, but when these numbers are kept, we keep guidelines alongside them. After over ten years in the industry I am a bit old-school and am used to the numbers. |
| | The process of figuring out these importance, that's decided upon between both the decision-makers and the stakeholders. |

**Table 5.9:** Insights From Practitioners on Criteria Considered for All Projects

| Practitioner (P) | Insights |
|---|---|
| P1 | Well, the reliability of the vendor and the financials are considered important regardless of the RFP. Besides the solution they provide, you're also looking for the sustainability or the stability of the vendor... Other than that, maybe the completeness or the consistency in the solution is also important. |
| P3 | It depends of course. Because each project in its domain is almost unique. But security is almost considered for all projects. Furthermore, whether the solution can be integrated into our current system... Whether training is provided or not and what is the service level which we will get after implementing... Also the system and language, because we have two languages ... so these are a few criteria available in all types of tenders. |

### User Roles and Responsibilities

The practitioners were further asked about the user roles and responsibilities, along with their views on transparency and bias and its handling within the system. In these discussions, the usefulness of utilizing MCDM methods such as AHP, especially due to the subjectivity that is part of the criteria and vendor evaluation, was brought forward. Transparency within the working groups (consisting of the stakeholders and decision-makers) was also highlighted. Consequently, it is essential that the decisions and scorings remain transparent to the stakeholders, even if they might not favor the outcomes for various reasons. The importance of transparency in these decisions is underscored by requirements such as conflict of interest disclosures for large contracts. This emphasizes the need for clear separation of duties and information transparency within the system. Anonymising the vendors was another suggestion to reduce the bias. Also, the importance of seeing conflicts and inconsistencies is proposed as useful to let the decision-makers discuss and disclose any uncertainties. Lastly, while transparency is considered important, not overdoing transparency is deemed important as too much transparency can result in biased or influenced decisions.

**Table 5.10:** Insights From Practitioners on User Roles and Responsibilities

| Practitioner (P) | Insights |
|---|---|
| P1 | ...practically by the subjectivity is part of the evaluation... AHP etc., may also help to remove remove the bias, not only calculating or summing up the responses given. |
| P2 | So ideally you've got a working group of, let's say, half a dozen or whatever it is. They will have full transparency to everything because if you don't, you cause too many problems. The process needs to be transparent to the key stakeholders. They may not like the outcome. There may be a lot of background politics getting played, but it has to be transparent. |
| | ...when you do this, depending on the size of organization, type complexity, especially if it's government, people have to disclose information, such as if they've got shares and company a manager or a family trust and and all of those sort of things. Especially with with the very big government contract. Perhaps you have to get auditors in sometimes and audit the decisions and make sure the conflicts of interest are disclosed and that sort of thing. And that's why having this segregation of duties built into the system is is key now when the working groups lands its decision. |
| P3 | See for me which I see is a bad thing even like because I come from a small country so we almost know everyone and every company. So I'm getting 30 proposal from my 30 companies. And me being biased to one or two. It's there even if I don't want to put it. Yeah, it's very even. I'll say no. I'll be, like, unbiased for all the all all of them. And I'll treat them equally. So the best approach will be to have to evaluate these companies or these tenders and proposals without knowing which company I am reviewing, like giving the scores and then going to know the company. |
| | And some users are like on average the users are giving seven and one user is giving one or zero. So maybe this person will ask like why you give him zero if it's something on average or everybody's giving. So no it's the same thing. But if somebody is out of line or unique they will ask him like, maybe he sees something which others are not seeing, so that's pointed out. Other than that, no, it's taken as it is. |
| | ...if we are picking too much transplant between all of us, I can become biased and influenced in my decision. |

### Tool Effectiveness and Usability

#### Intuitiveness

There was one explicit comment on the intuitiveness of the tool. The practitioner found the tool to look easy to navigate and use, but found the main-screens (dashboard) to be a bit confusing. They believed this might be due to them not being familiar with it and some terminology they did not get introduced to.

**Table 5.11:** Insights From Practitioners on Tool Effectiveness and Usability

| Practitioner (P) | Insights |
|---|---|
| P3 | It looks easy! But I got kind of lost in the main-screen. So maybe there there are some terminology I didn't know, yeah. I think if we see the main screen, it looked a bit confusing, but I think it's easy. It's like I'm not familiar with it so other than that, I think it's easy. |

#### Features Evaluation and Practitioner Wishes

Practitioners addressed the interesting and most valuable features of the decision-support tool, highlighting those that they found particularly beneficial. These features are listed in no particular order in Table 5.12 and are detailed in Table 5.13.

**Table 5.12:** Practitioners' View on Features in the Decision-Support Tool

| Current Features | Desired Additional Features |
|---|---|
| Checking and highlighting inconsistencies in the assigned scores | Looking into alternative ways to evaluate than using pair-wise comparison |
| Deciding on the decision-makers or evaluators for the project | Breaking criteria into sub-criteria |
| Assigning criteria that will be used | Spider Webs or look at more graph alternatives for visualizing various data |
| The way evaluation of the criteria priorities are done | Showing the criteria importance gathered from decision-makers |
| Use of both numbers and Fuzzy definitions | Having a summarized page summarizing all vendor details |
| Visualization of the ranking and scores | Having a built-in way of creating RFPs, with AI support |
| Separation of duties | Looking at ways AI can be used for data extraction and error management |
| Controlling access to information | A more comprehensive admin center for managing access and responsibilities |
| Admin center | Update dashboard to be more comprehensive and data-driven |
| Stage screens guiding the users through their processes | Dashboard updates for decision-makers to see graphs and numbers |
| Ingestion of RFPs and VPs into a workflow | Allow for binary scoring |
| Not knowing who the other DMs are | Having a guideline of what the various scores mean |
| Dashboard of all projects | Anonymize the vendors |
| | Customizability for various organizations |

In addition, practitioners were asked about the features or functionalities they felt were missing or would have liked to see incorporated into the tool. These desired features are also listed in no particular order in Table 5.12 and can be found detailed in Table 5.14, reflecting practitioner feedback and suggestions for potential enhancements.

### Overall Impact

Lastly, looking at the overall impact of the tool, including practitioner satisfaction with the proposed tool, whether they would like to see such a tool in their process and organization, and its usefulness in its current state, before finishing by addressing essential features that should be present in all such tools.

Two of the practitioners (P1 and P3) find the tool very promising and believe it will help a lot while conducting their evaluations. The last practitioner (P2) finds the tool to appear fine, but emphasize that their processes consider a lot of off-system tasks that needs to be further addressed to make the tool effective in their case, but that such a tool would significantly impact their processes.

The first practitioner (P1) found the tool to assist and improve the process a lot as scores and all evaluation steps are stored and conducted within the system, and all the necessary data around the solutions, proposals, etc., were there. Fur-

**Table 5.13:** Insights From Practitioners on the Tools Most Valuable Features

| **Practitioner (P)** | **Insights** |
|---|---|
| P1 | It's important to also check the inconsistencies. Is is very, very important... |
| | For example, decision on the decision-makers or the evaluators, and also the criteria that you will be evaluating the priorities when they are compared with each other, though those are very important during the evaluation. |
| | There's always uncertainty in the procedure, so you should keep it like it is (scoring using Fuzzy) |
| | There should be some ranking so that you can see your evaluation... I would use Spider Web kind of graph instead of bar chart, but such use of visualizations are really good. |
| P2 | And that comes back to one of the ones on the system around who sees what what administration center is set up to manage views and permissions and also enforcing the separation of duties. So So if Bob is a final decider, Bob may not have permission to rank security criteria... those that issue the check should not be the ones that make the decision... having this separation of duties built into the system is key... |
| | That's why having this admin center to manage views permissions, access segregation of duties, roles and responsibilities. Obviously you can't build it all into this prototype, but making it a point to say any system needs to have a good admin function admin center. |
| | Something along the lines that you've done with stage screens, sort of a wizard type process guide them through all has merit. |
| | The administration center, the role based access, the permissions, the segregation of duties. Controlling access to information that's an absolute must have. |
| | The second one is this ingestion of RFP's, RFT's and responses and some level of smarts around it within the workflow. |
| P3 | ...But let's say in a sense I'm not knowing the details like this answer is coming from which one? I'm just scoring them and in the end process if there is a conflict, there should be like a percentage of the conflict. If it's a small conflict, I think it should be ignored. If it is a major conflict, then it should be resolved without me knowing the conflict is coming from which side... If the windows are putting their details and I'm getting all the conflicts and how to resolve it, that's a good way to to look in the tool. |
| | This is a good part to have to have a dashboard of all the projects and what's going on in this project. What's the what they have reached. But yeah, it can be easy and it will be used only by a few people... |
| | For some questions, I'll need the numbers to be more specific. They are following this protocol and this protocol is affecting this, so maybe it will be a mix of the numbers, question answers and having this first thing. |

thermore, they also highlighted the need for customizability for making it feasible. The third practitioner (P3) found the tool to support them a lot in their processes despite some small things. They found it as a good decision-support tool, so much they suggested it should be commercialized.

All the practitioners agree on the tool being fine (or better) in its current state, but have various inputs on which parts they would like to see to make it perfect. Two of the practitioners wanted to assign a score to the current state of the tool out of six, where P1 gave it a four, P2 did not wish to assign a score as it was in a prototype state and their processes were too reliant on off-system activities, while the last practitioner P3 provided it a score of six. Table 5.15 shows the score distribution.

Lastly, some of the features that is to be deemed as essential in all such tools

**Table 5.14:** Insights From Practitioners on Features They Would Have Liked To See

| Practitioner (P) | Insights |
|---|---|
| P1 | ...you have to evaluate and consider proposals pair-wise which may not be easy... you may add some ranking or some scaling for each of the criteria or evaluations, having the possibility to score one vendor against the criteria in isolation. |
| | Adding sub criteria for the criteria is also important. For example, as we discussed, security may have some criteria and the other things so that you can cover all the parts of the tender. |
| | There should be some ranking so that you can see your evaluation. And I would use Spider Web graph kind of graph or some others instead of bar chart... So having a summarized page there that would be really, really interesting. |
| | Do you mention or show the weights that you have gathered from the decision makers? I think that can be useful. |
| P2 | Whether I know you may not have got that built in, but being able to have a more standard format RFP being able to ingest it using various forms of artificial intelligence... Also, you need to bring the workflow keeping some off-system processes put a little bit more rigor around that use AI within the tool for data extraction, field matching, error management and handling, pointing out quality errors and things like that. |
| | That's why having this admin center to manage views permissions, access separation of duties, roles and responsibilities. Obviously you can't build it all into this prototype, but making it a point to say any system needs to have a good admin function, admin center. |
| P3 | ...there are some terminology which, for example, I didn't know. |
| | It's good to have a summary. Like for example, kind of a chart or something. The numbering system like, because in in real life, in ANONYMIZED, there are on the in the same time maybe hundreds of projects going on, so you need kind of in the dashboard, from let's say 100 projects, 50 have reached this stage and 50 at this stage and 50 are overdue. Do you then like these agents have done nothing? A big summary in the dashboard, like what's going on because in the end of the day the decision maker in the we call it the purchasing department, each and every day he will see his dashboard and see like, what's going on and which area they should focus more and try to finish the project... So this dashboard or this page is mostly for the management part, yeah, but for a decision maker, he needs a dashboard like seeing graphs and numbers. These graphs and numbers. And he can make the decisions. |
| | In some cases, A or B there is. There is nothing in between. There is no score. It's like if there is score I'll give him 0 or 10. |
| | ...usually when these numbers are kept, we keep a guideline like, there is a guideline. |
| | Like I don't see which company is this. It shouldn't be mentioned in the proposal. OK, I will see the proposal itself, but nothing regarding the company itself. |

**Table 5.15:** Score Evaluation of Tool in Current Form

| Practitioner | Score out of 6 |
|---|---|
| P1 | 4 |
| P2 | N/A |
| P3 | 6 |

are side-by-side view of scoring and documents, and having all the different files in the system. Additionally, having an administration center to manage views and permissions and enforcing the separation of duties should be in all such systems,

**Table 5.16:** Insights From Practitioners on the Overall Impact of the Tool

| Practitioner (P) | Insights |
|---|---|
| P1 | This is something that will assist the process and improve the process a lot, so I would like to use it because you keep all the scores, you keep all the evaluation steps and you have all the necessary data around those kind of things like itself and solutions, etc... Very helpful. |
| | Upon the things we have talked about, missing points and customizability, I would say four. |
| | The features are important to the evaluation criteria are important, so the things should be adaptable or parametric. Maybe for each case and for each organization. That part could be customizable as much as possible. That's the most important and core point for making it feasible. |
| | Well, it sounds very promising indeed and it will help a lot while evaluating the solutions or the proposals. |
| P2 | Not at this stage. Thanks for asking. It all appears fine. Have a real think about the off-system things that are gonna be needed to make the tool effective. |
| P3 | To be honest, I was very interested when I was watching the presentation like to be honest, the tool looks good from all the perspectives. I think I'll say it's good, yeah it's good... It will support a lot, to be honest, it will support a lot. Like as I mentioned currently we are not having tools like each person is allowed to have their own Excel sheets and documents and and do the process as they like. Having a tool like this and having access to the what they call the PDF and the question in a similar way is a good way to submit my final answer. |
| | This thing, six. Very good. It's a great thing, to be honest. Some small things as I mentioned, but they're nothing just to fix them. It's a good decision-making tool. I think you should go and commercialize it. |
| | To be honest, the tool looks good from all perspectives... |

including such various roles. Moreover, a dashboard should be there for the users and also being able to anonymize the vendors depending on the projects should be possible.

**Table 5.17:** Insights From Practitioners on Features To Be Deemed As Essentials

| Practitioner (P) | Insights |
|---|---|
| P1 | For pair-wise comparison I would prefer side-by-side... also having all different files in the system. |
| P2 | The administration center, the role based access, the permissions, the segregation of duties, their controlling access to information that's an absolute must have... The second one is this ingestion of RFP's and responses and some level of smarts around it within the the workflow. |
| P3 | As we discussed, the dashboard should be there so that the person, the employee itself who is evaluating and the decision maker can see everything. And the one thing that we can add it to make the company anonymous, depending on the project, for example, you see, OK, this project is not required to be anonymous... But to have to have that kind of option in the system to make them anonymous and not make them anonymous is. |

This second round of practitioner evaluation marks the conclusion of the research's second iteration. Throughout this chapter, feedback from practitioners on the low-fidelity user interface design and the proposed tool process flow has been presented, followed by an in-depth examination of their perspectives on the finalized tool. Additionally, this evaluation has provided insights into the practices and

needs of various practitioners.

# Chapter 6

# Results

This chapter includes an analysis of NFRs as selection criteria, the enhancement of the ICR framework by Rani et al. [1] through the incorporation of Fuzzy Set Theory, and concludes with the key takeaways from the evaluation of the developed decision support tool. These findings are structured into three main sections: *NFRs as Selection Criteria*, which covers the systematic literature review on NFRs; *Fuzzy Scaling*, addressing the application and enhancement of the ICR framework with Fuzzy Set Theory; and *Decision-Support Tool*, highlighting insights from the tool's evaluation phases. In the subsequent Discussion chapter (Chapter 7), these results will be aligned with the research questions and literature gaps outlined in earlier chapters (Sections 1.4 and 2.4).

## 6.1 NFRs as Selection Criteria

The first result of this thesis relates to the SLR of NFRs as selection criteria in the context of software vendor selection. This review detailed the top 8 most used NFRs out of the selected papers and proposed a tailored set of definitions for this context, aiming to enhance both the effectiveness and efficiency of vendor selection processes by providing a robust set of selection criteria, specifically NFRs, that can help keystone companies and decision-makers make more informed decisions.

### 6.1.1 Background

Initially, to gain a clearer understanding of the current research landscape within software ecosystems, a review of various SLRs on software ecosystems was undertaken. This review aimed to identify whether these studies have already addressed the issues relevant to this research. The outcome of this review determined that no existing SLRs have adequately covered the current state of the art and state of the practice regarding selection criteria, particularly emphasizing NFRs, in the context of software vendor selection within software ecosystems.

### 6.1.2 Methodology

Following the methodology outlined by Keele et al. [67], an SLR was employed as the method for data collection. A subset of four key databases was chosen for the search out of those recommended by Keele et al. [67]; *Scopus*, *IEEE Xplore*, *ACM Digital Library*, and *SpringerLink*. These databases are highly suitable for an SLR centered on NFRs and selection criteria within software ecosystem decision-making. Their broad scope encompasses computer science, engineering, and business, providing access to a wide array of reputable, peer-reviewed publications. This extensive coverage supports a multidisciplinary approach, which is crucial given the varied aspects of software ecosystems. By drawing from their rich collections of journals, conference proceedings, and technical literature, the review gains depth and credibility, enabling a thorough exploration of scholarly insights in this specialized area.

These databases use advanced search queries to identify research papers. The search query had to be slightly modified for each database to accommodate variations in syntax and logic. The following search query was used as a foundational template:

*(TITLE-ABS-KEY ("software ecosystem" OR "SECO" OR "software outsourcing") OR TITLE-ABS-KEY ("decision making" OR "decision-making") OR TITLE-ABS-KEY ("vendor analysis" OR "vendor selection" OR "service provider") AND TITLE-ABS-KEY ("quality attributes" OR "selection criteria" OR "decision criteria" OR "non-functional requirements" OR "NFRs") AND TITLE-ABS-KEY (software)) AND PUBYEAR > 2012 AND PUBYEAR < 2024*

Filters, dictated by the exclusion criteria, were applied to certain databases where allowed. The applied filters included:

**Table 6.1:** Filters Applied during database search

| Database | Filters applied |
|---|---|
| SpringerLink | - |
| Scopus | EXCLUDE: Book chapter |
| IEEE Xplore | Conferences, Journals, Standards, 2013-2023 |
| ACM Digital Library | 2013-2023 |

Upon extraction of the studies, a selection protocol, comprising a set of inclusion and exclusion criteria, was employed to evaluate papers, proving especially crucial during the third stage of the SLR where two reviewers independently assessed the papers. This protocol was especially useful in extracting relevant data from the studies. The details of the selection protocol are available in Appendix E.

Five steps were carried out during the selection of primary sources:

**Step 1** Collection of all papers
**Step 2** Removal based on titles and protocol

**Step 3** Removal based on abstracts and conclusions
**Step 4** Full-text evaluation
**Step 5** Paper quality assessment

All Steps 1 through 4 are detailed in Appendix E, while Step 5 is highlighted here. Figure 6.1 highlights the included papers throughout the primary sources selection. Out of the initial 45510 papers, 1382 papers underwent some form of review, while 31 were included through *Step 4 - full-text evaluation*. It is worth highlighting that the evaluation conducted in *Step 3 - Removal based on abstracts and conclusions* was carried out by two reviewers in separation using the selection protocol to mitigate bias. Upon completion of this step, the results were discussed among the reviewers, and with the Kappa Cohen Statistic [68] evaluation, the reviewers showed *substantial agreement* with a *k-value* of 0.70.

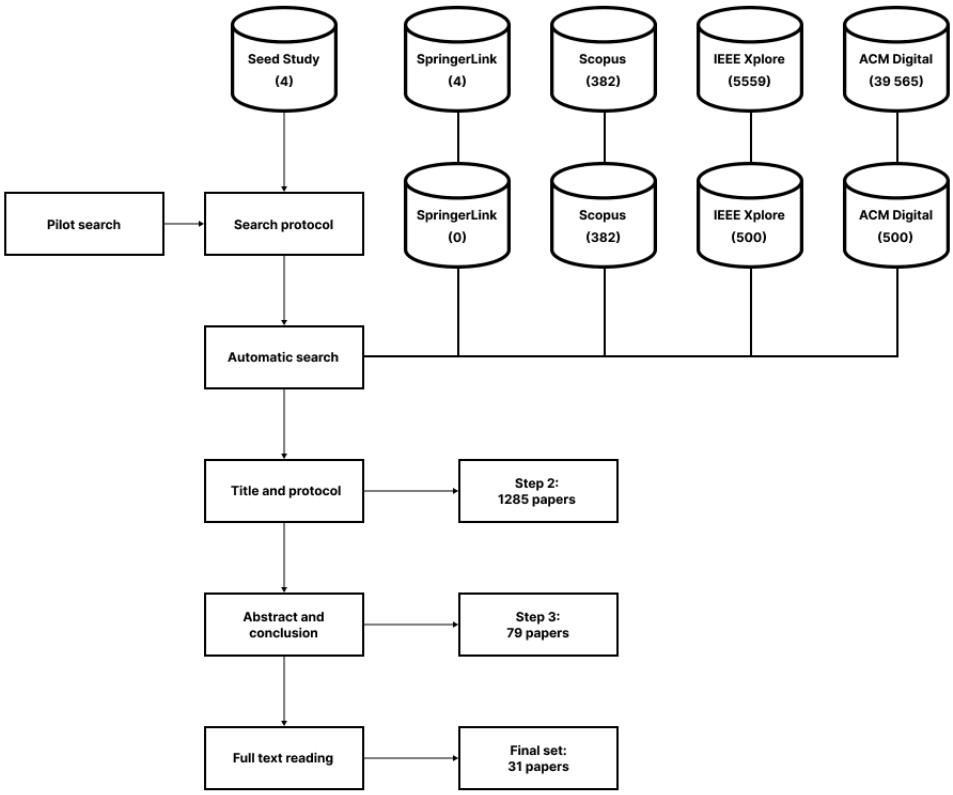


**Figure 6.1:** SLR Process

Lastly, upon selection of the 31 papers, a final step was added; *Step 5 - Paper quality assessment*. The evaluation of the papers was conducted following the approach proposed by Bertolino et al. [69] in their work on cloud testing. Their use of the "QualityScore" metric proved to be valuable in identifying limitations

within the studies. Consequently, this approach not only helped determine which papers are more reliable but also shed light on the current state of the literature in the specific field.

The "QualityScore" is calculated as the sum of individual scores, as outlined in Equation 6.1. Individual scores are assigned on a scale from 0 to 1, where 0 signifies that the paper does not meet the criterion, 0.5 indicates partial satisfaction, and 1 indicates full compliance. This evaluation is performed for each criterion, and the scores for all criteria are then aggregated to obtain a total "QualityScore" (Equation 6.1) out of the maximum attainable score, *7*. Upon assessment completion, zero studies were placed in the *poor* category, 18 were categorized as *good*, while 13 were deemed to be of *excellent* quality. The distribution of studies for the quality assessment can be seen in Figure 6.2 and Figure 6.3.

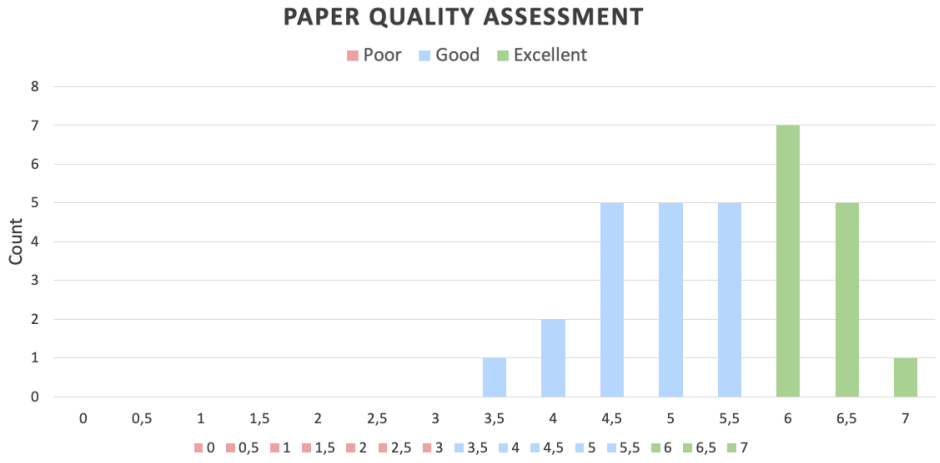


**Figure 6.2:** Paper Quality Assessment Score Distribution


**Figure 6.3:** Paper Quality Assessment Category Distribution

It is worth noting that this assessment was carried out by a single author, which represents a limitation. However, this approach was adopted to provide additional insights into the credibility of the research findings.

$$QualityScore = \sum_{k=1}^{7} I_k \tag{6.1}$$

The quality assessment checklist is highlighted in Table 6.2.

| # | Item |
|---|---|
| 1 | Is the problem of the study clearly defined? |
| 2 | Is the contribution of the study clearly defined? |
| 3 | Is the methodology clearly communicated? |
| 4 | Are the results clearly communicated? |
| 5 | Are limitations and future directions clearly stated? |
| 6 | Do they clearly state the application domain? |
| 7 | Is the focus on software ecosystems clearly defined? |

**Table 6.2:** Quality Assessment Checklist

| Answer Scores for the Items |
|---|
| No = 0; Partially = 0.5; Yes = 1 |

**Table 6.3:** Answer Scores for Items

### 6.1.3 Findings

There is a significant challenge met when considering NFRs as selection criteria [10, 51, 53]. One difficulty linked to NFRs lies in their interdependencies and the mutual influence on one another [53]. Therefore, they cannot be evaluated in isolation but must be considered collectively [53]. Due to their complexity, companies often decide to not formally specify these quality requirements, but rather treat them during the development process [70]. Excluding consideration of NFRs during the decision-making process can be fatal for both the product and the ecosystem, and Lytra et al. [9] emphasize that while many ADDs for instance concern functionalities of the system, the quality attributes are often among the most important decision drivers.

#### Top 8 NFRs

From all the papers of the review, 61 NFRs are extracted in various frequencies and from various domains. Eight of the NFRs are addressed in four or more papers and therefore included as requirements of interest in this thesis.

These requirements, listed in no particular order, are:

1. Scalability
2. Performance
3. Reliability
4. Security
5. Usability
6. Maintainability
7. Portability
8. Availability

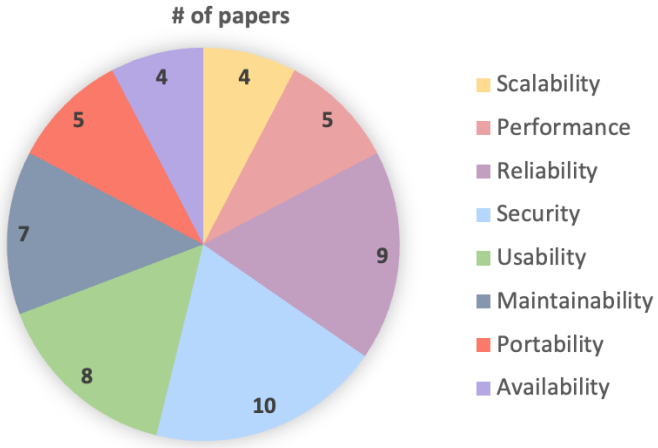


**Figure 6.4:** NFR Frequency Addressed

Why is it necessary to propose such a set of criteria and address their definitions? The complexity of software ecosystem structures highlights the importance of properly understanding and managing requirements engineering, especially focusing on quality requirements to ensure smooth collaboration. Yet, the subjective nature of decision-making [9], especially when including NFRs as selection criteria [10], often leads to their neglect. This highlights the urgent need to deepen the understanding of how NFRs are used as selection criteria, aiming at finding patterns that help decision-makers handle these requirements more effectively. While quality standards like ISO/IEC 25010 provide a basic definition of NFRs, they only partially aid decision-makers, leaving much dependent on their subjective interpretation. Therefore, reviewing the literature and these quality standards to develop definitions for NFRs specifically suited to the software ecosystem context seems a logical step forward.

### Interpretations

This section provides a comprehensive overview of how requirements are covered in the literature, assessing the strengths and weaknesses of their current definitions. It concludes by proposing a definitive version of these requirements' defin-

itions, specifically tailored as selection criteria within the context of software vendor selection in software ecosystems. Extensive details on the findings from the review, such as studies and domain each requirement is highlighted in, the extent to which each of the studies did their work, and a detailed walk-through of the reasoning behind the proposed definitions along with any proposed definitions within the studies can be seen in Appendix F. Furthermore, inspecting the domains and motivations of the studies under review is also presented in that Appendix. However, to summarize; the collection of studies included in this SLR spans a bunch of domains, including fields such as the Industry Automation Sector, Health Care software, Automotive Industry, Cloud environments, ERP systems, Smart Cities, the KDE ecosystem, and software development and software ecosystems more in general, to mention a few. The overall distribution of studies within which domains are presented in Figure 6.5.

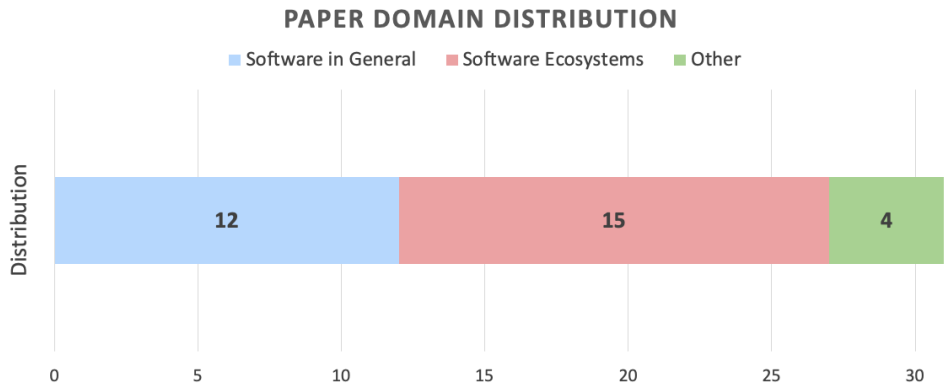


**Figure 6.5:** Paper Domain Distribution

Ultimately, the goal is to consolidate these insights to propose a set of requirements and definitions. The objective of these is to establish a robust foundation for stakeholders to provide decision-makers with during the vendor selection process.

***Availability***

The first requirement of interest is *availability*. Among the four studies that address this topic, three — namely those by Belinda et al. [29], Kumar et al. [71], and Sharma et al. [55] — provides definitions for this requirement . The concept of availability is also detailed in the ISO/IEC 25010:2023 standard. Lytra et al. [9], however, do not offer a definition or explanation of availability, leaving its interpretation to the understanding of decision-makers.

Availability can assume varied meanings depending on the application field and perspective, as indicated by the literature findings. Nevertheless, it appears that the existing discussions fail to fully capture what availability entails in the context of software vendor selection within a software ecosystem. Therefore, it is proposed to formulate a clear, specific definition and recommendation to encour-

age a consistent and unambiguous understanding of availability when selecting software vendors.

In the context of a software ecosystem, availability should be considered not only in terms of the software product provided by the vendor but also in terms of the vendor's availability itself. Drawing on the ISO/IEC 25010:2023 standard, the insights from Belinda et al. [29], Sharma et al. [55], and personal interpretations, a comprehensive definition of availability is suggested as follows:

*Software availability in an ecosystem denotes the product's and vendor's ability to ensure continuous access and functionality. It combines the software's reliability with vendor support to minimize downtime, ensuring the product is operational and accessible as needed, according to defined performance standards and agreements.*

Existing definitions often overlook the broader scope of interpretations relevant to software vendor selection. It is, therefore, crucial to evaluate this criterion within the dual context of both vendor and product availability to provide decision-makers with a holistic view, rather than limiting the focus to software product availability alone, as is often the case. By embracing this comprehensive approach, it is reasonable to believe that decision-makers can enhance their ability to choose vendors that meet the reliability and operational standards vital for successful participation in software ecosystems.

***Scalability***

The second requirement of interest is *scalability*. Surprisingly, none of the four studies reviewed which addressed scalability provided a definition of the criteria or its sub-criteria, leaving a gap in the literature-based understanding of this concept. However, the ISO/IEC 25010:2023 standard identifies scalability as a sub-criterion of flexibility.

Similarly to availability, it is critical to evaluate both the technical capabilities of the software product and the vendor's capacity to support organizational growth when selecting software vendors within ecosystems. Therefore, drawing on the ISO/IEC 25010:2023 standard and additional personal interpretations, a comprehensive definition of scalability is proposed:

*Scalability in a software ecosystem highlights the product's and vendor's capability to support growth, emphasizing the ability to adapt to increasing demands and business needs. It encompasses the software's customization flexibility and global expansion capabilities, ensuring that solutions not only evolve with but also support performance consistency across different regions.*

Because of the limited discussions on scalability in the existing literature and especially in the context of software vendor selection, there is a clear need for a redefined concept. While the ISO/IEC 25010:2023 standard adequately covers aspects of product scalability, it overlooks the capabilities of vendors. The redefinition suggested here includes considerations of both product and vendor capabil-

ities related to scalability, particularly tailored to the needs of software vendor selection within software ecosystems. This refined approach is intended to enhance decision-makers ability to make well-informed choices that facilitate successful scalability within software ecosystems.

***Portability***

The third requirement of interest is *portability*. Although five of the papers reviewed address this criterion, only three — namely those by Jansen et al. [72], Belinda et al. [29], and Amorim et al. [4] — provide specific definitions. Additionally, portability is defined within the ISO/IEC 25010:2023 standard.

Current definitions from the literature and the ISO/IEC 25010:2023 standard offer valuable insights into the concept of portability. By synthesizing these definitions, a more robust and comprehensive definition can be formulated, for the benefit of decision-makers in software vendor selection.

Drawing on insights from the ISO/IEC 25010:2023 standard, Jansen et al. [72], Belinda et al. [29], Amorim et al. [4] and personal interpretations, an overall definition of *portability* is suggested as follows:

*Portability in a software ecosystem encapsulates the software's flexibility to adapt and function across diverse computing environments without extensive modification. It merges the principles of platform independence, data migration efficiency, and interoperability, ensuring seamless operation and integration, regardless of the underlying technology.*

While the existing definitions found in literature and the ISO/IEC 25010:2023 standard offer valuable insights into the portability requirement, the proposed definition provide a nuanced view of portability that emphasizes platform independence, data migration efficiency, and interoperability — all of which are crucial factors in evaluating portability. With this refined understanding, decision-makers are better equipped to assess the portability of both vendors and their software products, facilitating more informed selection within software ecosystems.

***Performance***

The fourth requirement of interest is *performance*. Among the studies reviewed, five addressed performance, but only two — specifically the works of Belinda et al. [29] and Ameller et al. [73] — along with the ISO/IEC 25010:2023 standard, provided definitions for this criterion.

The definitions of performance from the literature and the ISO/IEC 25010:2023 standard offer valuable insights. However, it is suggested that these definitions should be expanded to include vendor performance, an aspect which is currently overlooked. Based on the ISO/IEC 25010:2023 standard, the proposed definitions of Belinda et al. [29] and Ameller et al. [73], along with personal interpretations, an overall definition of performance is proposed to be:

*Performance in the context of software ecosystems involve evaluating the software's efficiency in resource utilization and operational effectiveness, including scalability,*

*response times, and reliability. Concurrently, it requires assessing the vendor's ability to meet service level agreements, innovate, and adapt to technological advancements, ensuring they provide robust support and continuous improvement.*

Although the definitions of software product performance in the literature and the ISO/IEC 25010:2023 standard provide valuable insights, a redefined interpretation is suggested. This revised definition integrates elements from the existing descriptions and extends them to include the performance context of vendors, aiming to provide decision-makers with a deeper and more nuanced understanding of performance, particularly in relation to software vendor selection. The proposal emphasizes key principles such as efficiency in resource utilization, operational effectiveness, scalability, response times, and reliability, all considered critical for thoroughly assessing software products and vendors. This enhanced perspective equips decision-makers with a robust framework to effectively evaluate both vendors and their products, addressing the evolving needs and complexities of software ecosystems.

***Maintainability***

The fifth requirement of interest is *maintainability*. Seven studies address maintainability, but only three — namely, the two works of Amorim et al. [4, 74] and Belinda et al. [29] — provide distinct definitions. These definitions frame maintainability both in isolation and as a sub-criterion of flexibility. Lastly, the ISO/IEC 25010:2023 standard also provides its definition.

Although the definitions offered by the literature and the ISO/IEC 25010:2023 standard establish a solid foundation for maintainability, there is a benefit for decision-makers of synthesizing the strengths of these definitions into a singular definition tailored for the software vendor selection context. This enhanced definition of maintainability is proposed as follows:

*Maintainability in a software ecosystem reflects the software's and vendor's capability to efficiently and effectively modify the product to correct faults, enhance performance, or adapt to changing requirements.*

This revised definition aims to provide decision-makers with a clearer, more actionable understanding of maintainability, crucial for evaluating both software products and vendors. It specifically aids in assessing how well vendors and their software can address maintenance tasks such as correcting faults, improving performance, and adapting to changing needs based on detailed scenarios from the RFPs and VPs. By offering a more targeted definition, the intention is to simplify the decision-making process, enabling a more effective assessment of maintainability in the context of software ecosystems.

***Usability***

The sixth requirement of interest is *usability*. Usability is discussed in eight of the studies reviewed, but only four provide definitions, alongside the ISO/IEC

25010:2023 standard. These studies include the works of Kocak et al. [75], Amorim et al. [74], Belinda et al. [29], and Chazette et al. [76]. One thing worth noting is that the ISO/IEC 25010:2023 standard also addresses usability, though under the renamed term of *interaction capability*.

The definitions of usability found in existing literature and the ISO/IEC 25010:2023 standard offer valuable insights. Nonetheless, integrating these definitions could offer decision-makers a more precise and clear understanding, particularly valuable in the context of software vendor selection.

Drawing from the ISO/IEC 25010:2023 standard, alongside the insights from Kocak et al. [75], Amorim et al. [74], Belinda et al. [29], and Chazette et al. [76], a comprehensive definition of usability is proposed:

*Usability in a software ecosystem denotes the software's capacity to provide an intuitive, learnable, and efficient interface that enables users to achieve their objectives with satisfaction in a specified context of use. It encompasses the system's adaptability to user needs, facilitating engagement and reducing error rates by guiding users towards correct actions and minimizing misunderstandings.*

By synthesizing elements from existing definitions, this enhanced description of usability helps clarify its critical components, emphasizing its importance in assessing software products. This approach ensures that decision-makers are equipped with the necessary criteria to evaluate whether vendors' products meet the desired usability standards effectively.

***Security***

The seventh requirement of interest is *security*. This criterion is the most frequently addressed in the literature, with ten studies discussing it, yet only two of them offer a definition. The ISO/IEC 25010:2023 standard also provides a definition.

The definitions currently available in the literature are primarily based on the earlier version of the ISO 25010. These definitions, together with the ISO/IEC 25010:2023 standard, align well with the needs identified. However, it is suggested that these definitions are expanded to include the vendor's capability to secure information and data, especially concerning information intended for ecosystem participants. Additionally, a refinement in the communication of these definitions are proposed.

Drawing from the ISO/IEC 25010:2023 standard, alongside insights from Dayanandan et al. [77] and Amorim et al. [4], a comprehensive definition of security is recommended:

*Security in a software ecosystem is defined by the software's and vendor's ability to safeguard information and data, ensuring access is strictly provided according to the users' authorization levels and effectively protecting against unauthorized access and malicious threats.*

While the definitions of security in existing literature and the ISO/IEC 25010:2023 provide valuable insights, a slight modification to include the vendor's capability to secure information and data is suggested. By doing so, decision-makers will need to carefully examine the vendor more thoroughly, not just the software product they offer. This could involve steering clear of vendors with a history of mishandling critical organizational data, which could potentially harm the ecosystem either as a whole or in terms of competitiveness.

***Reliability***

The eighth and final requirement of interest from the selection of studies is reliability. This attribute is highlighted in nine papers, with five providing definitions, complemented by the ISO/IEC 25010:2023 standard. Researchers such as Lytra et al. [9], Amorim et al. [74], Ameller et al. [73], Kumar et al. [71], and Kocak et al. [75] all emphasize the importance of reliability along with a proposed definition.

While the definitions of reliability in the existing literature and the ISO/IEC 25010:2023 standard offer valuable insights into reliability in software systems, they tend to overlook considerations of vendor reliability. To address this gap, it is advisable to include vendor reliability in the definition, integrating the strengths of the various definitions found in the literature.

Drawing from the ISO/IEC 25010:2023 standard and the comprehensive insights from the studies mentioned, a refined definition of reliability is suggested:

*Reliability in a software ecosystem refers to the software's and vendor's capability to consistently perform specified functions accurately and without failure under stated conditions for a defined duration.*

This enhanced definition of reliability both synthesizes the key elements from existing definitions but also incorporates the aspect of vendor reliability. This addition is crucial, as it underscores the importance of vendor dependability in fulfilling commitments within the ecosystem, not only the software product. By acknowledging both product and vendor reliability, the revised definition aims to provide a clearer, more comprehensive understanding of reliability, ensuring successful collaborations within the ecosystem.

**Summary**

While there are multiple studies present in the literature considering non-functional requirements and proposing a definition of these, most of them fail to capture the whole perspective. These studies focus mainly on the software product quality, not on assessing the vendor's capability to meet some of these criteria. All the criteria are therefore redefined, mostly due to the lack of vendor considerations, something which in this study is deemed essential when assessing NFRs as selection criteria.

## 6.2 Fuzzy Scaling

The second result of this thesis is the adaptation and expansion of the Inconsistency and Conflict Removal (ICR) framework, as proposed by Rani et al. [1], to handle decision-making under conditions of uncertainty. To address the challenges associated with imprecise and uncertain data within the AHP framework, numerous studies have successfully implemented the linguistic scale of judgment and fuzzy numbers for performance evaluation, also known as Fuzzy Set Theory (FST), as further detailed in Section 2.2.3. Such implementations have significantly improved the decision-making methods' adaptability and usability. Consequently, it was a logical step to enhance the state-of-the-art framework developed by Rani and colleagues by integrating an additional layer that facilitates the application of FST.

The method utilized to incorporate Fuzzy has been a comprehensive review of current literature to understand how such a technique can be most effectively incorporated into the AHP without going on behalf of the features of the ICR framework. The works of Ayhan [56] served particularly useful for the incorporation as it contained a step-by-step walk-through of how to conduct the Fuzzy calculations in AHP with a supplier selection example. This study did, however, not have any code implementation associated with it making the steps serve as a baseline for the code implementation. This code can be seen in the GitHub repository[1] and its process flow is detailed in Section 4.4.4.

Fuzzy Set Theory is applied to the layers for assigning and calculating scores within the ICR framework (Figure 6.6). This means that the ICR method within the ICR framework can operate normally without modifications. The difference is that instead of calculating weights and scoring vendors on the server side using the Saaty values, this calculation is done using Fuzzy values. This is possible due to the use of a separate algorithm that operates server-side, as further detailed in Section 4.4.4. The triangulation format and *Saaty Scale-Fuzzy Triangular Scale* mapping, as introduced in Section 2.2.3 are utilized to achieve this. By utilizing such a format, the user can interact with Fuzzy values, the ICR check can still operate with Saaty values and the calculations of criteria weights and vendor capabilities are made utilizing the Fuzzy with just a couple of server-side value mappings.

## 6.3 Decision-Support Tool

The third and final result is related to the tool that was developed on top of the ICR framework as illustrated in Figure 6.6 along with practitioner feedback. The tool development is detailed in Chapter 4 and the evaluation of both the first and second phases are detailed in Chapter 5. In this section, the focus will be on summarizing the practitioner feedback related to the tool evaluation and its

[1] `https://github.com/MACS490-TMM/Tool`

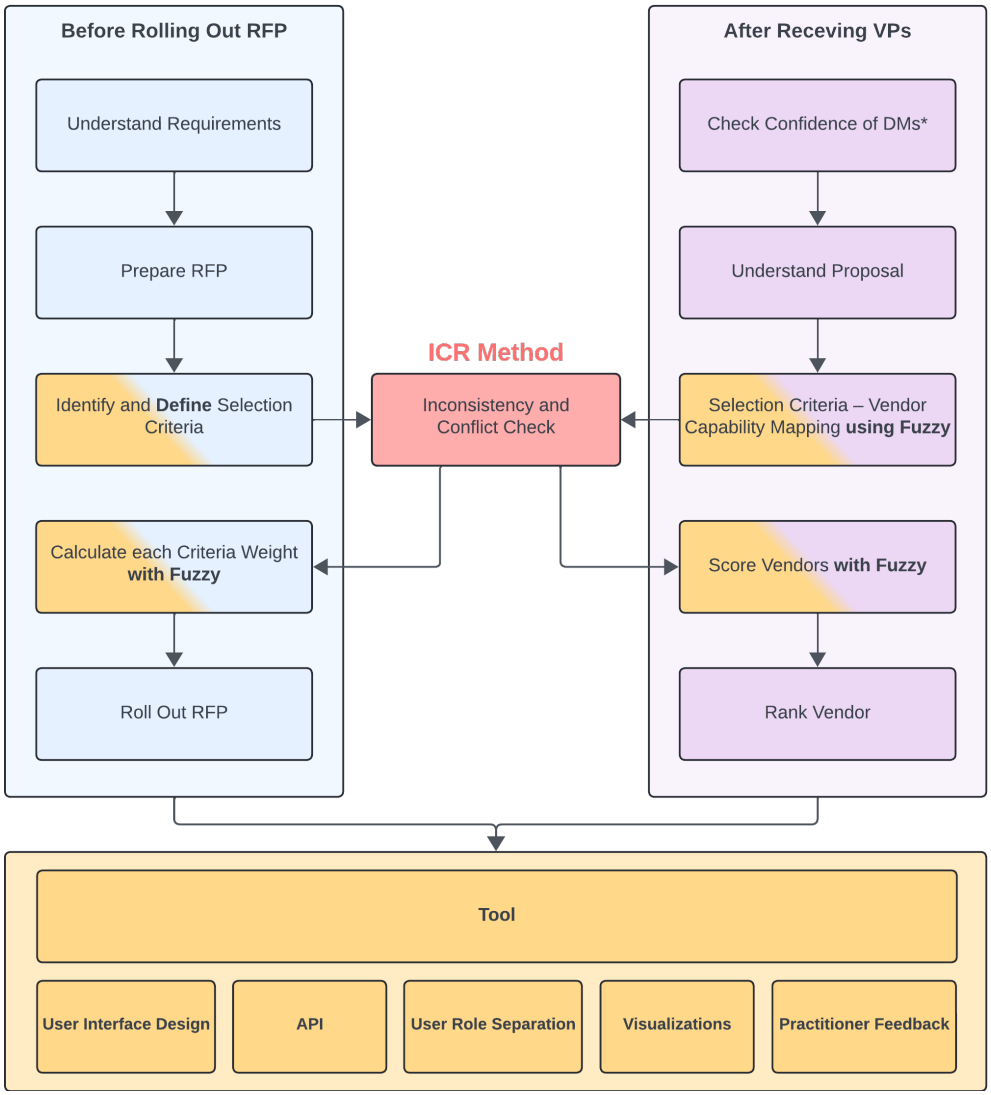


**Figure 6.6:** ICR Framework with the Three Contributions

usability. These results will serve as the foundation for the final assessment of the tool which is further detailed in Section 7.1.3.

The features that were highlighted by the practitioners as most interesting or valuable, along with additional features the practitioners would have liked to see in the decision-support tool are addressed in Table 5.12.

A shared consensus among the practitioners was that the decision-making process is an ad-hoc and tedious process mostly conducted through manual labor in tools like Excel or even on paper. Despite the flexibility, learnability, and simplicity of setup within tools like Excel, it is considered tedious and not preferred by practitioners. The proposed decision-support tool, however, is deemed to meet multiple of the decision-makers' wishes for such tools and alignment with current

decision-making processes. It even presents features not present in other tools in an effective way. Details on the practitioners' feedback, both regarding the tool's strengths, weaknesses and suggested modifications can be seen in Section 5.2.2. A summary of the decision-makers' overall feedback on the tool and its usability is presented next:

*'They are doing just like you...'*

*'This is something that will assist the process and improve the process a lot, so I would like to use it because you keep all the scores, you keep all the evaluation steps and you have all the necessary data around those kind of things like itself and solutions, etc... Very helpful.'*

*'...it sounds very promising indeed and it will help a lot while evaluating the solutions or the proposals.'*

*'...It all appears fine.'*

*'...to be honest, the tool looks good from all the perspectives... It will support a lot, to be honest... Like as I mentioned currently we are not having tools like each person is allowed to have their own Excel sheets and documents and and do the process as they like. Having a tool like this and having access to the what they call the PDF and the question in a similar way is a good way to submit my final answer...It's a good decision-making tool.* ***I think you should go and commercialize it****...'*

# Chapter 7

# Discussion

In this chapter, the results and the implication of these findings will be addressed and used along with the literature findings to address the research questions and how this research have addressed them (Figure 7.1). Thereafter, any limitations and threats to validity will be addressed.

## 7.1 Research Questions

This research work is conducted to address three primary research questions, along with a series of sub-questions, as outlined in Section 1.4. This section explains how each research question was addressed, either through insights obtained from the literature or via the proposed implementation.

### 7.1.1 RQ 1: How are different NFRs addressed as selection criteria for software vendor selection in software ecosystems?

As seen from the preliminary looks on selection criteria, it became evident that well-defined selection criteria are crucial for informed and successful decision-making [26, 27]. However, the subjective nature of NFRs often leads to their neglect [9], creating a gap in understanding and interpreting them in different contexts.

A systematic literature review was conducted to investigate how NFRs are interpreted for software analysis and selection (Section 1.3). The findings emphasize that NFRs has been used for quality assessment of software products on the broader level, however, limited literature is available for their interpretation as selection criteria for vendor analysis and selection. As addressed in Sections 2.2.2 and 6.1.3, various studies look at the use of NFRs both in general and as selection criteria. While there are multiple studies addressing these NFRs in software ecosystems (Figure 6.5), they fall short in their ability to generalize their criteria and definitions as well as mostly assess only the software product, not including the vendors. To the best of the author's knowledge, no studies have addressed NFRs

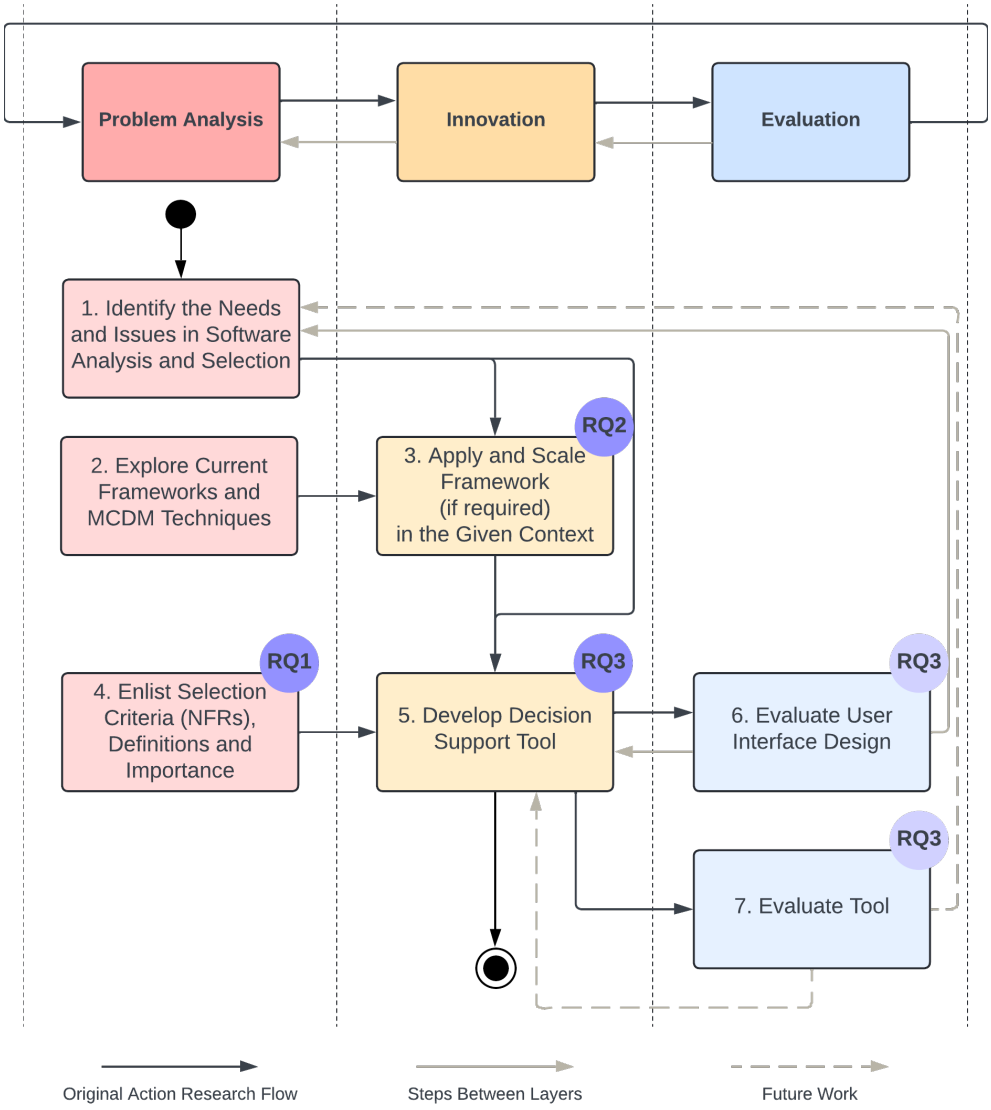


**Figure 7.1:** Action Research with Contributions

as selection criteria for software vendor selection in software ecosystems, where both the software product and vendor are assessed.

### RQ1 a: Which NFRs are utilized as selection criteria?

Out of the set of 61 NFRs appearing at varying frequencies in the SLR , eight were identified and chosen as the foundational set. These eight were selected because they were discussed in four or more of the final 31 papers. A threshold of four was chosen as a natural cutoff, as the next most frequent occurrence after four was only two. Furthermore, the inclusion of these criteria in the ISO/IEC 25010:2023

standard, along with their proposed definitions and the domains they originated from, further reinforced their role as selection criteria within this context.

These eight requirements ended up being: *Scalability*, *Performance*, *Reliability*, *Security*, *Usability*, *Maintainability*, *Portability*, and *Availability*.

Although these criteria are frequently mentioned in the studies reviewed, it's important to recognize that they might not fit every project or ecosystem due to unique characteristics. However, the main point is their broad applicability. These criteria have been successfully applied across various domains, making them ideal as a generalized, out-of-the-box option for the vendor selection processes. Stakeholders can adapt this baseline set, adding or omitting criteria as needed, to suit most contexts effectively.

Looking ahead, while the proposed criteria are considered a robust set for vendor selection, there are several directions for future research. First, it is crucial to validate these criteria through various case studies and discussion with practitioners to evaluate their applicability and generalizability. This was initially planned for in this study, however, due to limited access to practitioners, this step was not prioritized. Second, there is potential to expand this list if additional NFRs are identified as critical for selection. Given that this is the first comprehensive list of criteria specifically tailored for software vendor selection in ecosystems, and considering the methodology used to select the studies reviewed, it is reasonable to believe that not all relevant studies were included. Consequently, this list may evolve as more studies are considered.

**RQ1 b: How can NFRs be interpreted to better assist decision-makers in this particular context?**

Upon identifying how NFRs are addressed as selection criteria and proposing a out-of-the-box set of eight requirements for software vendor selection in software ecosystems, understanding how these criteria could be interpreted to better assist decision-makers in the given context was deemed essential. Approximately 40% (12) of the studies examined were broadly related to general software and outsourcing, while around 50% (15) were directly tied to software ecosystems (Figure 6.5). Understanding the application of these criteria both in general software contexts and specifically in software vendor selection was critical. This need was underscored by the tendency to overlook NFRs due to the subjective nature of decision-making [9]. Proposing a set of clear definitions or explanations for these requirements was seen as a way to mitigate some of the subjectivity [10], particularly concerning the intended meaning of each requirement [29]. Furthermore, a review of the literature revealed that these criteria are often interpreted differently across various fields (Appendix F). Therefore, synthesizing the best insights from these findings and addressing the lack of specific definitions, for example concerning vendor considerations led to the proposition of new, tailored definitions for the context of software vendor selection in software ecosystems (Section 6.1.3). This approach aims at providing decision-makers with clearer criteria, thereby facilit-

ating more objective and informed decisions. The proposed approach to refining how NFRs are interpreted specifically aims to enhance decision-making in the context of software vendor selection within ecosystems. This involves broadening the scope of criteria definitions to include not only the software product but also the vendor itself. Additionally, minor revisions to existing definitions were suggested as part of the solution to better support decision-makers in this specific setting.

Looking back, there are some limitations to bringing vendor considerations into a single statement for the affected criteria. While this approach draws attention to vendor capabilities, it can lead to confusion and conflicting assessments within a single criterion. For instance, if the vendor's software performs well but the vendor itself has limitations, two distinct scores might be required during the assessment of a single criterion, which is not ideal. A potential solution would be to divide these criteria into two separate sub-criteria under a common parent criterion [29], each with its own level of importance. This modification would allow for a clearer and more accurate evaluation of both software performance and vendor capabilities.

These definitions of the non-functional selection criteria pave new paths for considering NFRs during the vendor selection process. Currently, the criteria and their definitions are based solely on literature and the ISO/IEC 25010:2023 standard, and have not yet been validated by practitioners. Incorporating practitioner feedback in the next evaluation phase will be essential to confirm their relevance and importance.

### 7.1.2 RQ 2: Which criteria weighting strategies are effectively utilized within this context and how can these be integrated into the ICR framework?

As evidenced by the literature (Section 2.2.3), it is apparent that traditional pairwise comparison methods like the Analytical Hierarchy Process (AHP), which uses Saaty values ranging from 1 to 9, do not adequately account for decision-making under conditions of uncertainty. To overcome the challenges associated with imprecise and uncertain data, the integration of a practical strategy within the AHP framework has been effective - namely Fuzzy Set Theory (FST). This strategy incorporates a linguistic scale of judgment and fuzzy numbers for performance assessment. As evidenced in various software and supplier selection studies, this approach effectively combines the strengths of both methodologies while mitigating their respective weaknesses [40–48]. The effectiveness of incorporating FST as a weighting strategy is further supported by findings from the SLR, where five studies explicitly discuss its successful application in weighting NFRs as selection criteria (Section 2.2.3) [51–55]. Moreover, the combination of FST with AHP has been shown to improve the adaptability and usability of AHP in diverse case studies [46, 49, 50].

The demonstrated success of FST in enhancing decision-making under uncertainty, along with its role in improving the adaptability and usability of AHP, has

led to its adoption as the core strategy for integration into the Inconsistency and Conflict Removal (ICR) framework developed by Rani et al. This integration maintains the integrity of the ICR method while enhancing its effectiveness, allowing decision-makers to assign values to criteria more intuitively (Section 6.2). This is achieved by enabling partial membership and utilizing textual importance levels, which align closely with human cognitive processes, thus reducing cognitive load for decision-makers.

A detailed step-by-step guide on applying Fuzzy AHP in supplier selection was adopted to implement the Fuzzy scaling within the ICR framework. This guide outlines all necessary steps, leaving the remaining task being the code implementation (Section 4.4.4). By storing the Saaty values in the system and mapping these to the fuzzy definitions on the client-side, and to the Fuzzy Triangulation Scale on the algorithm side (Table 2.6), the ICR method operates seamlessly and without modifications. The fuzzy scaling were made to the ICR framework in the steps of; *calculating criteria weight*, *criteria-vendor capability mapping*, and *vendor scoring*, as highlighted in Figure 6.6. To ensure successful implementation, the steps in the guide was performed in the implemented algorithm, serving identical results.

A consideration during the project was whether to allow decision-makers to assign scores to criteria and vendors individually rather than using a pairwise approach. This idea was also suggested by a practitioner during the walk-through of their preferences during tool evaluation. However, the inherent interdependencies among NFRs, which affect each other significantly, make isolated evaluation challenging, as noted in the work of Saadatmand et al. [53]. Consequently, given that the ICR framework employs the AHP which relies on pairwise scoring, this approach was not included in the current project scope. Nevertheless, this aspect should be further explored in future research to potentially simplify the decision-making process.

### 7.1.3 RQ 3: How can decision-making processes be streamlined to enhance usability?

Current decision-making processes are complex, ad-hoc, and for the most part manual, as highlighted by literature and practitioner feedback (Section 2.2.1 and Chapter 5.2.2). Despite employing dedicated personnel for decision-making roles, the process lacks transparency and remains subjective [1, 9]. Various methods and frameworks like the ICR framework by Rani et al. [1] have been developed to support decision-makers. However, a common limitation persists across these approaches: they often overlook or fall short of addressing the adaptability and usability necessary for integration within the processes of keystone companies and their decision-makers. Specifically the ICR framework, while it increases the adaptability and usability of the AHP, it requires users to be familiar with its codebase limiting its use to those with technical expertise which compels decision-makers to revert to ad-hoc methods. Furthermore, the framework still rely on multiple off-system processes such as manual document handling and score tracking, to

mention a few.

In this thesis, a decision-support tool, consisting of a user-interface and logic connecting the user-interface to the data handling and processing was developed to address the limitations of adaptability and usability inherent in the ICR framework. This tool proposes to guide decision-makers through their journey from project setup to the final decision, following the structure proposed by the ICR framework and aligned with practitioner processes. The tool's details was presented in Section 4. By providing an all-in-one solution that systematically organizes the necessary information for decision-making at the time the user needs it, the tool aims to enhance the usability of current methods. This approach is also wished for by various studies with beliefs that it will enhance the adaptability and usability of current proposed methods [1, 57].

While the tool initially was planned to focus more towards the use of visualizations due to its positive impact on the decision-making process [57], it became evident that such focus was premature, due to the foundation needed to effectively utilize these visualizations was missing. The decision-support tool, equipped with various features, was subsequently evaluated by practitioners. Although certain aspects were identified for improvement (Section 6.3), the general consensus among practitioners was positive. They believed that the implementation of such a tool would significantly improve the usability of current methods and greatly assist them in their decision-making processes:

*'This is something that will assist the process and improve the process a lot, so I would like to use it because you keep all the scores, you keep all the evaluation steps and you have all the necessary data around those kind of things like itself and solutions, etc... Very helpful.'*

*'...it sounds very promising indeed and it will help a lot while evaluating the solutions or the proposals.'*

*'...to be honest, the tool looks good from all the perspectives... It will support a lot, to be honest... Like as I mentioned currently we are not having tools like each person is allowed to have their own Excel sheets and documents and and do the process as they like. Having a tool like this and having access to the what they call the PDF and the question in a similar way is a good way to submit my final answer...It's a good decision-making tool. I think you should go and commercialize it...'*

## 7.2 Limitations and Threats to Validity

In this section, some of the limitations faced are addressed. Thereafter, some of the threats to validity will be highlighted and discussed.

### 7.2.1 Limitations

The limitations addressed are related to the strengths and volume of literature, the practitioner availability and lastly, the development team size.

#### Literature

The literature review showed there are noticeable gaps in the research, especially regarding how NFRs are used as selection criteria in software ecosystems and how decision-making tools are evaluated. These gaps likely limited the understanding and steered the direction of this study. Because there's little research on tools specifically for software vendor selection, the scope had to widened to include general decision-making tools. While this was necessary, it might have resulted in missing some important details specific to software vendor selection. Also, due to limited information on NFRs, insights had to be withdrawn from various fields and creatively interpret the data to make it relevant to this study.

#### Practitioner Availability

Another limitation of the study was related to the availability of practitioners. The decision-making roles targeted for this study are often held by highly skilled individuals who occupy senior positions within companies and have many responsibilities. This made it challenging in two ways. First, it was difficult to engage these busy professionals for the study due to their packed schedules. Second, those who agreed to participate could only commit to a limited number of sessions, which were primarily used for evaluations. While their contributions were invaluable and provided crucial insights, having them more involved throughout the tool's development process would have likely enhanced the outcome and made the research align even more with the foundation of action research. The ICR framework developers were closely involved throughout the development process and were able to represent the practitioners to a certain degree based on their knowledge.

#### Development Team Size

The last limitation that will be addressed is the limited development team size. As the development team only consisted of one developer, certain features had to be prioritized and unfortunately leave out some desired functionalities. For instance, this limitation led to the exclusion of crucial phases such as testing, deployment, and maintenance from the SDLC, which could affect the tool's reliability and scalability in a real-world setting and simplified the manual checks.

### 7.2.2 Threats to Validity

Two types of threats to validity needs to be addressed in this study - namely internal and external validity.

#### Internal Validity

To enhance internal validity, several measures were implemented. A pre-recorded video presentation of the tool walk-through was created to ensure that all practitioners received the same foundational knowledge about the tool. This approach,

together with a standardized interview guide, was designed to minimize variability in responses and ensure a consistent base of knowledge during the interview process. However, the semi-structured nature of the interviews might pose a threat to internal validity. As insights and lessons were drawn from each interview, knowledge on where to direct the conversation evolved, potentially leading to inconsistencies in the data collected across sessions.

Another potential threat to internal validity stems from the decision-making experience of the participant group. All practitioners involved had extensive experience in decision-making. While their expertise was valuable for evaluating the tool against their comprehensive backgrounds, this homogeneity may limit the applicability of the findings to less experienced decision-makers. The perspectives of newer decision-makers might differ significantly, particularly regarding the usability and adaptability of the tool.

**External Validity**

While various measures were taken to strengthen the external validity, despite limited availability of practitioners, the set of practitioners contained a diverse set of practitioners both in terms of geographic location, organization type and processional responsibilities. This was in turn done to enhance the generalizability of the results. Despite these measures, the study suffers from presenting the tool as a video and show various features again upon request of the participants, instead of having the practitioners use the tool.

Efforts to strengthen external validity were prioritized given the limited availability of practitioners. The participant group was intentionally diverse, encompassing a variety of geographic locations, organization types, and professional responsibilities. This diversity was aimed at enhancing the generalizability of the results across different contexts. Despite these measures, the study encounters limitations that may affect its external validity. One significant limitation is the way the tool is presented. The tool was demonstrated via video and specific features were shown again at the participants' request rather than allowing practitioners to interact with the tool directly. This was done due to limited time availability of participants and as addressed in Section 7.2.2 to increase internal validity, but the method of presentation may not accurately reflect the tool's usability and functionality in a real-world setting, potentially impacting the applicability of the findings. Furthermore, the small sample size poses a threat to the external validity of the findings, potentially limiting the generalizability of the results to a wider population.

# Chapter 8

# Conclusion and Future Work

The conclusion and future work chapter is the final chapter of the thesis, aiming at wrapping up the work that has been conducted and propose future work that should find place.

## 8.1 Conclusion

Software ecosystems involve outsourcing the development of parts or entire systems to vendors, enabling companies to collaborate in addressing challenges and achieving shared goals. In such ecosystems, the dependency on other entities heightens the risk of being affected by their failures. Consequently, choosing the right vendor for integration into the ecosystem is critical but challenging, requiring highly skilled personnel. Nonetheless, the selection process is often complex, ad-hoc, and manual, characterized by its subjective nature and lack of transparency. To accommodate this, various decision-making methods and frameworks are employed, although these strategies frequently fall short in addressing decision-making under uncertainty and in their adaptability and usability for decision-makers in their processes.

This thesis makes contributions in three areas. Firstly, the thesis identifies and interprets NFRs as selection criteria within the context of software vendor selection in ecosystems, highlighting the most commonly used NFRs and their definitions. Secondly, the thesis enhances the state-of-the-art ICR framework to enable decision-makers to effectively manage imprecise data, subjective judgments, and multiple conflicting objectives through the Fuzzy implementation. Lastly, the thesis proposed a developed end-to-end decision-support tool based on the scaled ICR framework that guides decision-makers through their processes in a systematic, automated, and streamlined manner. Such a tool was proposed as one of the possible solutions to how decision-making processes can be streamlined to enhance usability.

The tool underwent evaluation by practitioners with extensive decision-making experience, holding various background, providing invaluable feedback on its effectiveness and usability. Among other things, they identified the most valuable

features and those they wished to see implemented. Additionally, their insights into their own processes have enriched the understanding and will serve central in informing future iterations of the tool. The significance of the proposed tool to streamline and enhance the usability in decision-making process is further validated by the expert practitioners along with the an accepted publication of the low-fidelity design and tool process flow, as detailed in Section 1.5.

## 8.2 Future Work

Various directions should be taken in future research, both in terms of NFRs role as selection criteria, and regarding the decision-support tool. Some of which was highlighted in Section 7.1. The proposed set of criteria, while robust, requires validation through practical application. Future studies should focus on conducting case studies or similar and engaging with practitioners to assess the applicability and generalizability of the proposed criteria. Furthermore, as this is the first comprehensive list of NFRs tailored as selection criteria for software vendor selection in software ecosystems, there is a potential to expand this list. Future research should explore the inclusion of additional NFRs that may be critical for selection. Given the methodology used to select the reviewed studies (Section 6.1), it is possible that not all relevant studies were included, suggesting that this criteria list may evolve as further studies are reviewed. Additionally, it may be beneficial to explore whether the criteria definitions should encompass both software product assessment and vendor capabilities within a the same definition.

Regarding criteria weighting strategies, future work should investigate whether and how criteria and vendors can be evaluated independently (not pair-wise) within the ICR framework.

Concerning the decision-support tool, several enhancements are possible. A key upgrade involves transitioning from simple JSON file storage to using a database, which would enhance data management and scalability. Addressing the features highlighted by practitioners (Table 5.12) in subsequent tool iterations will better align the tool with practitioner needs. Additionally, future research should continue exploring how visualizations can be effectively utilized and incorporated into the tool and decision-makers' processes, enhancing the clarity and impact of the data presented. Completing the remaining phases of the proposed SDLC (Figure 4.1) is also crucial. Finally, it is crucial to let practitioners actively use and interact with the tool in their processes. This hands-on application will help identify additional needs and issues which requires attention.

## 8.3 Concluding Remarks

This thesis explored how software vendor selection in software ecosystems can be enhanced. Initially, it identified and interpreted non-functional requirements as selection criteria. Subsequently, the ICR framework was adapted to incorpor-

ate Fuzzy Set Theory, enable decision-makers to more effectively handle imprecise data, subjective judgments, and multiple conflicting objectives. Most importantly, the development and evaluation of a decision-support tool were detailed, which demonstrated its utility in streamlining the selection process. Future research could focus on expanding the tool's capabilities, exploring its application in diverse ecosystems, and further enhancing its integration with existing enterprise systems to maximize its effectiveness and reach.

The hope is that this thesis work will guide attention toward improving the complex processes of software vendor selection, ensuring that similar scandals as observed in the British Post Office scandal are not repeated.

# Bibliography


[1] A. Rani, D. Mishra and A. Omerovic, 'A framework for software vendor selection by applying inconsistency and conflict removal (icr) method,' *International Journal of System Assurance Engineering and Management*, Nov. 2023, ISSN: 0976-4348. DOI: 10.1007/s13198-023-02190-x. [Online]. Available: https://doi.org/10.1007/s13198-023-02190-x.

[2] A. Rani, D. Mishra and A. Omerovic, 'Multi-vendor software ecosystem: Challenges from company' perspective,' in *Information Systems and Technologies*, A. Rocha, H. Adeli, G. Dzemyda and F. Moreira, Eds., Cham: Springer International Publishing, 2022, pp. 382–393, ISBN: 978-3-031-04829-6.

[3] D. Alao, S. Okolie and O. Awodele, 'Software ecosystem: Features, benefits and challenges,' *International Journal of Advanced Computer Science and Applications*, vol. 4, Sep. 2013. DOI: 10.14569/IJACSA.2013.040833.

[4] S. Amorim, S. Andrade, J. Mcgregor, E. Almeida and C. Chavez, 'Tailoring the nfr framework for measuring software ecosystems health,' Jan. 2018. DOI: 10.17771/PUCRio.wer.inf2018-14.

[5] A. Rani, D. Mishra and A. Omerovic, 'Exploring and extending research in multi-vendor software ecosystem,' in *Innovations in Smart Cities Applications Volume 5*, M. Ben Ahmed, A. A. Boudhir, İ. R. Karaş, V. Jain and S. Mellouli, Eds., Cham: Springer International Publishing, 2022, pp. 379–391, ISBN: 978-3-030-94191-8.

[6] S. Jansen, A. Finkelstein and S. Brinkkemper, 'A sense of community: A research agenda for software ecosystems,' in *2009 31st International Conference on Software Engineering - Companion Volume*, 2009, pp. 187–190. DOI: 10.1109/ICSE-COMPANION.2009.5070978.

[7] E. Sparrow, *A Guide to Global Sourcing: Offshore Outsourcing and Other Global Delivery Models*. British Computer Society, 2004, ISBN: 9781906124250. [Online]. Available: https://books.google.no/books?id=Vr7EasqADjEC.

[8] K. Manikas, 'Revisiting software ecosystems research: A longitudinal literature study,' *Journal of Systems and Software*, vol. 117, pp. 84–103, 2016, ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2016.02.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121216000406.

[9] I. Lytra, G. Engelbrecht, D. Schall and U. Zdun, 'Reusable architectural decision models for quality-driven decision support: A case study from a smart cities software ecosystem,' in *2015 IEEE/ACM 3rd International Workshop on Software Engineering for Systems-of-Systems*, May 2015, pp. 37–43. DOI: 10.1109/SESoS.2015.14.

[10] T. Olsson, K. Wnuk and T. Gorschek, 'An empirical study on decision making for quality requirements,' *Journal of Systems and Software*, vol. 149, pp. 217–233, 2019, ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2018.12.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121218302668.

[11] I. Hunink, R. van Erk, S. Jansen and S. Brinkkemper, 'Industry taxonomy engineering: The case of the european software ecosystem,' in *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pp. 111–118.

[12] J. Verville and A. Halingten, 'A six-stage model of the buying process for erp software,' *Industrial Marketing Management*, vol. 32, no. 7, pp. 585–594, 2003, ISSN: 0019-8501.

[13] Last accessed 31 May 2024, Apr. 2024. [Online]. Available: https://corporate.postoffice.co.uk/en/horizon-scandal-pages/context.

[14] Last accessed 31 May 2024, May 2024. [Online]. Available: https://en.wikipedia.org/wiki/British_Post_Office_scandal.

[15] *Post Office Horizon scandal: Why hundreds were wrongly prosecuted*, May 2024, Last accessed 31 May 2024. [Online]. Available: https://www.bbc.com/news/business-56718036.

[16] C. Stokel-Walker, *What the hell is going on with the U.K. Post Office?*, Jan. 2024, Last accessed 31 May 2024. [Online]. Available: https://www.fastcompany.com/91017767/uk-post-office-explained.

[17] O. Hornstein, *Fujitsu valuation drops by* $1bnamidPostOfficescandalscrutiny$, Jan. 2024, Last accessed 31 May 2024. [Online]. Available: https://www.uktech.news/saas/fujitsu-valuation-drops-post-office-contracts-scrutiny-20240112.

[18] T. Baker, *Post Office scandal: £1bn set aside to fund compensation for victims - as govt vows to pursue Fujitsu if fault found*, Jan. 2024, Last accessed 31 May 2024. [Online]. Available: https://news.sky.com/story/post-office-scandal-1bn-set-aside-to-fund-compensation-for-victims-as-govt-vows-to-pursue-fujitsu-if-fault-found-13046172.

[19] S. Jansen, S. Peeters and S. Brinkkemper, 'Software ecosystems: From software product management to software platform management.,' in *IW-LCSP@ ICSOB*, 2013, pp. 5–18.

[20] H. Kobayashi and H. Osada, 'Strengthening of collaboration between it vendors and their customer through proposal-based sales,' in *2011 IEEE International Conference on Quality and Reliability*, Sep. 2011, pp. 556–560. DOI: 10.1109/ICQR.2011.6031601.

[21] V. Boucharas, S. Jansen and S. Brinkkemper, 'Formalizing software ecosystem modeling,' in *Proceedings of the 1st International Workshop on Open Component Ecosystems*, ser. IWOCE '09, Amsterdam, The Netherlands: Association for Computing Machinery, 2009, pp. 41–50, ISBN: 9781605586779. DOI: 10.1145/1595800.1595807. [Online]. Available: https://doi.org/10.1145/1595800.1595807.

[22] E. Yu and S. Deng, 'Understanding software ecosystems: A strategic modeling approach,' in *Iwseco-2011 software ecosystems 2011. proceedings of the third international workshop on software ecosystems. brussels, belgium*, 2011, pp. 65–76.

[23] V. Bali, S. Bali, D. Gaur, S. Rani and R. Kumar, 'Commercial-off-the shelf vendor selection: A multi-criteria decision-making approach using intuitionistic fuzzy sets and topsis,' *Operational Research in Engineering Sciences: Theory and Applications*, vol. 6, no. 2, Sep. 2023. [Online]. Available: https://oresta.org/menu-script/index.php/oresta/article/view/571.

[24] A. S. Jadhav and R. M. Sonar, 'Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach,' *Journal of Systems and Software*, vol. 84, no. 8, pp. 1394–1407, 2011, ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2011.03.034. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016412121100077X.

[25] A. Plugge and H. Bouwman, 'Tensions in global it multisourcing arrangements: Examining the barriers to attaining common value creation,' *Journal of Global Information Technology Management*, vol. 21, no. 4, pp. 262–281, 2018. DOI: 10.1080/1097198X.2018.1536595. eprint: https://doi.org/10.1080/1097198X.2018.1536595. [Online]. Available: https://doi.org/10.1080/1097198X.2018.1536595.

[26] T. Lima, R. Santos and C. Werner, 'Seco-am: An approach for maintenance of it architecture in software ecosystems,' in *2020 XLVI Latin American Computing Conference (CLEI)*, Oct. 2020, pp. 242–251. DOI: 10.1109/CLEI52000.2020.00035.

[27] T. Olsson and K. Wnuk, 'Qreme – quality requirements management model for supporting decision-making,' in Mar. 2018, pp. 173–188, ISBN: 978-3-319-77242-4. DOI: 10.1007/978-3-319-77243-1_11.

[28] B. Schwichtenberg and G. Engels, 'Secoarc: A framework for architecting healthy software ecosystems,' in *Software Architecture*, H. Muccini, P. Avgeriou, B. Buhnova, J. Camara, M. Caporuscio, M. Franzago, A. Koziolek, P. Scandurra, C. Trubiani, D. Weyns and U. Zdun, Eds., Cham: Springer International Publishing, 2020, pp. 95–106, ISBN: 978-3-030-59155-7.

[29] B. I. Belinda, A. A. Emmanuel, N. Solomon and A. B. Kayode, 'Evaluating software quality attributes using analytic hierarchy process (ahp),' *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, 2021. DOI: 10.14569/IJACSA.2021.0120321. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2021.0120321.

[30] R. Kamalakannan, C. Ramesh, M. Shunmugasundaram, P. Sivakumar and A. Mohamed, 'Evaluvation and selection of suppliers using topsis,' *Materials Today: Proceedings*, vol. 33, pp. 2771–2773, 2020, International Conference on Nanotechnology: Ideas, Innovation and Industries, ISSN: 2214-7853. DOI: https://doi.org/10.1016/j.matpr.2020.02.105. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785320308518.

[31] D. Mairiza, D. Zowghi and V. Gervasi, 'Utilizing topsis: A multi criteria decision analysis technique for non-functional requirements conflicts,' in Apr. 2014, vol. 432, pp. 31–44, ISBN: 978-3-662-43609-7. DOI: 10.1007/978-3-662-43610-3_3.

[32] E. N. Madi, J. M. Garibaldi and C. Wagner, 'An exploration of issues and limitations in current methods of topsis and fuzzy topsis,' in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, pp. 2098–2105. DOI: 10.1109/FUZZ-IEEE.2016.7737950.

[33] T. L. Saaty, 'Making and validating complex decisions with the AHP/ANP,' *Journal of Systems Science and Systems Engineering*, vol. 14, no. 1, pp. 1–36, Mar. 2005.

[34] H. Taherdoost and M. Madanchian, 'Analytic network process (anp) method: A comprehensive review of applications, advantages, and limitations,' *Journal of Data Science and Intelligent Systems*, vol. 1, May 2023. DOI: 10.47852/bonviewJDSIS3202885.

[35] R. Saaty, 'The analytic hierarchy process—what it is and how it is used,' *Mathematical Modelling*, vol. 9, no. 3, pp. 161–176, 1987, ISSN: 0270-0255. DOI: https://doi.org/10.1016/0270-0255(87)90473-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0270025587904738.

[36] P. P. Roman Hruška and D. Babić, 'The use of ahp method for selection of supplier,' *Transport*, vol. 29, no. 2, pp. 195–203, 2014. DOI: 10.3846/16484142.2014.930928. eprint: https://doi.org/10.3846/16484142.2014.930928. [Online]. Available: https://doi.org/10.3846/16484142.2014.930928.

[37] M. C. Tam and V. Tummala, 'An application of the ahp in vendor selection of a telecommunications system,' *Omega*, vol. 29, no. 2, pp. 171–182, 2001, ISSN: 0305-0483. DOI: https://doi.org/10.1016/S0305-0483(00)00039-6. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305048300000396.

[38] F. Tahriri, M. R. Osman, A. Ali, R. Yusuff and A. Esfandiary, 'Ahp approach for supplier evaluation and selection in a steel manufacturing company,' eng, *Journal of Industrial Engineering and Management (JIEM)*, vol. 1, no. 2, pp. 54–76, 2008, ISSN: 2013-0953. DOI: 10.3926/jiem.v1n2.p54-76. [Online]. Available: https://hdl.handle.net/10419/188372.

[39] M. Dağdeviren, 'Decision making in equipment selection: An integrated approach with ahp and promethee,' *Journal of Intelligent Manufacturing*, vol. 19, no. 4, pp. 397–406, Jan. 2008. DOI: 10.1007/s10845-008-0091-7. [Online]. Available: http://dx.doi.org/10.1007/s10845-008-0091-7.

[40] C. E. Bozdağ, C. Kahraman and D. Ruan, 'Fuzzy group decision making for selection among computer integrated manufacturing systems,' *Computers in Industry*, vol. 51, no. 1, pp. 13–29, 2003, ISSN: 0166-3615. DOI: https://doi.org/10.1016/S0166-3615(03)00029-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361503000290.

[41] A. Calabrese, R. Costa and T. Menichini, 'Using fuzzy ahp to manage intellectual capital assets: An application to the ict service industry,' *Expert Systems with Applications*, vol. 40, no. 9, pp. 3747–3755, 2013, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.12.081. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741741201322X.

[42] C.-W. Chang, C.-R. Wu and H.-L. Lin, 'Applying fuzzy hierarchy multiple attributes to construct an expert decision making process,' *Expert Systems with Applications*, vol. 36, no. 4, pp. 7363–7368, 2009, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2008.09.026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417408006465.

[43] C.-W. Chang, C.-R. Wu and H.-L. Lin, 'Integrating fuzzy theory and hierarchy concepts to evaluate software quality,' *Software Quality Journal*, vol. 16, no. 2, pp. 263–276, Jun. 2008, ISSN: 0963-9314. DOI: 10.1007/s11219-007-9035-2. [Online]. Available: https://doi.org/10.1007/s11219-007-9035-2.

[44] T. Ertay, A. Kahveci and R. Tabanli, 'An integrated multi-criteria group decision-making approach to efficient supplier selection and clustering using fuzzy preference relations,' *Int. J. Computer Integrated Manufacturing*, vol. 24, pp. 1152–1167, Dec. 2011. DOI: 10.1080/0951192X.2011.615342.

[45] T. Le, A. Genovese and L. Koh, 'Using fahp to determine the criteria for partner's selection within a green supply chain,' *Journal of Manufacturing Technology Management*, vol. 23, pp. 25–55, Jan. 2012. DOI: 10.1108/17410381211196276.

[46] H.-Y. Lin, P.-Y. Hsu and G.-J. Sheen, 'A fuzzy-based decision-making procedure for data warehouse system selection,' *Expert Systems with Applications*, vol. 32, no. 3, pp. 939–953, 2007, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2006.01.031. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417406000509.

[47] M. Shamsuzzaman, S. Ura and E. Bohez, 'Applying linguistic criteria in fms selection: Fuzzy-set-ahp approach,' *Integrated Manufacturing Systems*, vol. 14, pp. 247–254, May 2003. DOI: 10.1108/09576060310463190.

[48] M. Zeydan, C. Çolpan and C. Çobanoğlu, 'A combined methodology for supplier selection and performance evaluation,' *Expert Systems with Applications*, vol. 38, no. 3, pp. 2741–2751, 2011, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2010.08.064. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417410008602.

[49] O. Kilincci and S. A. Onal, 'Fuzzy ahp approach for supplier selection in a washing machine company,' *Expert Systems with Applications*, vol. 38, no. 8, pp. 9656–9664, 2011, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2011.01.159. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417411001928.

[50] G. Secundo, D. Magarielli, E. Esposito and G. Passiante, 'Supporting decision-making in service supplier selection using a hybrid fuzzy extended ahp approach,' *Business Process Management Journal*, vol. 23, no. 1, pp. 196–222, Feb. 2017. DOI: 10.1108/bpmj-01-2016-0013. [Online]. Available: http://dx.doi.org/10.1108/bpmj-01-2016-0013.

[51] I. Lytra and U. Zdun, 'Supporting architectural decision making for systems-of-systems design under uncertainty,' in *Proceedings of the First International Workshop on Software Engineering for Systems-of-Systems*, ser. SESoS '13, Montpellier, France: Association for Computing Machinery, 2013, pp. 43–46, ISBN: 9781450320481. DOI: 10.1145/2489850.2489859. [Online]. Available: https://doi.org/10.1145/2489850.2489859.

[52] M. T. J. Ansari, F. A. Al-Zahrani, D. Pandey and A. Agrawal, 'A fuzzy topsis based analysis toward selection of effective security requirements engineering approach for trustworthy healthcare software development,' *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, Sep. 2020. DOI: 10.1186/s12911-020-01209-8. [Online]. Available: http://dx.doi.org/10.1186/s12911-020-01209-8.

[53] M. Saadatmand and S. Tahvili, 'A fuzzy decision support approach for model-based tradeoff analysis of non-functional requirements,' in *2015 12th International Conference on Information Technology - New Generations*, Apr. 2015, pp. 112–121. DOI: 10.1109/ITNG.2015.24.

[54] J. Mitchell, S. Rizvi and J. Ryoo, 'A fuzzy-logic approach for evaluating a cloud service provider,' in *2015 1st International Conference on Software Security and Assurance (ICSSA)*, 2015, pp. 19–24. DOI: 10.1109/ICSSA.2015.014.

[55] C. Sharma and S. K. Dubey, 'Reliability evaluation of software system using ahp and fuzzy topsis approach,' in *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, M. Pant, K. Deep, J. C. Bansal, A. Nagar and K. N. Das, Eds., Singapore: Springer Singapore, 2016, pp. 81–92, ISBN: 978-981-10-0451-3.

[56] M. B. Ayhan, *A fuzzy ahp approach for supplier selection problem: A case study in a gear motor company*, 2013. arXiv: 1311.2886 [cs.AI].

[57] C. P. Killen, J. Geraldi and A. Kock, 'The role of decision makers' use of visualizations in project portfolio decision making,' *International Journal of Project Management*, vol. 38, no. 5, pp. 267–277, 2020, ISSN: 0263-7863. DOI: https://doi.org/10.1016/j.ijproman.2020.04.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263786320300260.

[58] C. Gencer and D. Gürpinar, 'Analytic network process in supplier selection: A case study in an electronic firm,' *Applied Mathematical Modelling*, vol. 31, no. 11, pp. 2475–2486, 2007, ISSN: 0307-904X. DOI: https://doi.org/10.1016/j.apm.2006.10.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0307904X06002368.

[59] A. Dargi, A. Anjomshoae, M. R. Galankashi, A. Memari and M. B. M. Tap, 'Supplier selection: A fuzzy-anp approach,' *Procedia Computer Science*, vol. 31, pp. 691–700, 2014, 2nd International Conference on Information Technology and Quantitative Management, ITQM 2014, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2014.05.317. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050914004943.

[60] L. M. Padilla, S. H. Creem-Regehr, M. Hegarty and J. K. Stefanucci, 'Decision making with visualizations: A cognitive framework across disciplines,' *Cognitive Research: Principles and Implications*, vol. 3, no. 1, p. 29, Jul. 2018.

[61] K. Stølen, *Technology research explained design of software, architectures, methods, and technology in general*. Springer Nature, 2023.

[62] I. Solheim and K. Stølen, *Technology research explained*, 2007. [Online]. Available: https://sintef.brage.unit.no/sintef-xmlui/handle/11250/2387932.

[63] G. I. Susman and R. D. Evered, 'An assessment of the scientific merits of action research.,' *Administrative Science Quarterly*, vol. 23, pp. 582–603, 1978. [Online]. Available: https://api.semanticscholar.org/CorpusID:144445559.

[64] J. McGrath, *Groups: Interaction and Performance*. Prentice-Hall, 1984, ISBN: 9780133657005. [Online]. Available: https://books.google.no/books?id=4pzZAAAAMAAJ.

[65] Accessed 30 May 2024. [Online]. Available: https://aws.amazon.com/what-is/sdlc/.

[66] Jan. 2024. [Online]. Available: https://www.computer.org/resources/importance-of-software-design-is-important.

[67] S. Keele *et al.*, *Guidelines for performing systematic literature reviews in software engineering*, 2007.

[68] 2011. [Online]. Available: https://en.wikipedia.org/wiki/Cohen%27s_kappa.

[69] A. Bertolino, G. D. Angelis, M. Gallego, B. García, F. Gortázar, F. Lonetti and E. Marchetti, 'A systematic review on cloud testing,' *ACM Comput. Surv.*, vol. 52, no. 5, Sep. 2019, ISSN: 0360-0300. DOI: 10.1145/3331447. [Online]. Available: https://doi.org/10.1145/3331447.

[70] G. Valenca, C. F. Alves, V. Heimann, S. Jansen and S. Brinkkemper, 'Competition and collaboration in requirements engineering: A case study of an emerging software ecosystem,' in *IEEE 22nd International Requirements Engineering Conference, RE 2014, Karlskrona, Sweden, August 25-29, 2014*, T. Gorschek and R. R. Lutz, Eds., IEEE, 2014, pp. 384–393, ISBN: 978-1-4799-3033-3. DOI: 10.1109/RE.2014.6912289. [Online]. Available: http://dx.doi.org/10.1109/RE.2014.6912289.

[71] R. R. Kumar, M. Shameem, R. Khanam and C. Kumar, 'A hybrid evaluation framework for qos based service selection and ranking in cloud environment,' in *2018 15th IEEE India Council International Conference (INDICON)*, Dec. 2018, pp. 1–6. DOI: 10.1109/INDICON45594.2018.8987192.

[72] S. Jansen, 'How quality attributes of software platform architectures influence software ecosystems,' in *Proceedings of the 2013 International Workshop on Ecosystem Architectures*, ser. WEA 2013, Saint Petersburg, Russia: Association for Computing Machinery, 2013, pp. 6–10, ISBN: 9781450323147. DOI: 10.1145/2501585.2501587. [Online]. Available: https://doi.org/10.1145/2501585.2501587.

[73] D. Ameller, C. Ayala, J. Cabot and X. Franch, 'Non-functional requirements in architectural decision making,' *IEEE Software*, vol. 30, no. 2, pp. 61–67, 2013. DOI: 10.1109/MS.2012.176.

[74] S. d. S. Amorim, J. D. Mcgregor, E. S. d. Almeida and C. v. F. Garcia Chavez, 'Connecting non-functional requirements to open source ecosystems health,' in *Proceedings of the 16th Brazilian Symposium on Software Components, Architectures, and Reuse*, ser. SBCARS '22, Uberlandia, Brazil: Association for Computing Machinery, 2022, pp. 76–80, ISBN: 9781450397452. DOI: 10.1145/3559712.3559719. [Online]. Available: https://doi.org/10.1145/3559712.3559719.

[75] S. A. Koçak, G. I. Alptekin and A. B. Bener, 'Evaluation of software product quality attributes and environmental attributes using anp decision framework,' English, Cited by: 18, vol. 1216, 2014, pp. 37–44. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908292804&partnerID=40&md5=03d9e18dfe2eb69745df0031fc89be23.

[76] L. Chazette and K. Schneider, 'Explainability as a non-functional requirement: Challenges and recommendations,' *Requirements Engineering*, vol. 25, no. 4, pp. 493–514, Dec. 2020, ISSN: 1432-010X. DOI: 10.1007/s00766-020-00333-1. [Online]. Available: https://doi.org/10.1007/s00766-020-00333-1.

[77] U. Dayanandan and V. Kalimuthu, 'Software architectural quality assessment model for security analysis using fuzzy analytical hierarchy process (FAHP) method,' *3D Research*, vol. 9, no. 3, p. 31, Jul. 2018.

# Appendix A

# HCSE Paper

# A Preliminary User Interface for Software Vendor Analysis and Selection Tool

Tormod Mork Müller[1][0009−0005−6070−3714], Anshul Rani[2*][0000−0003−4500−290X], and Deepti Mishra[3][0000−0001−5144−3811]

[1] Norwegian Institute of Science and technology, Gjøvik, Norway
tormod.m.muller@ntnu.no
[2] Norwegian Institute of Science and technology, Gjøvik, Norway
anshul.rani@ntnu.no-corresponding author
[3] Norwegian Institute of Science and technology, Gjøvik, Norway
deepti.mishra@ntnu.no

**Abstract.** In software ecosystems, the decision-making process for vendor selection remains a significant challenge, often worsened by the lack of effective tools that leverage visualization to aid in these decisions. This study aims to bridge this gap by prototyping a tool and hence proposing a user interface for the same, based on a software vendor selection framework, enhancing decision-making through intuitive visualizations and a user-friendly interface. Our research methodology combined a semi-structured literature review, preliminary user interface development, and expert feedback to ensure the tool's practicality and effectiveness. The prototype testing of the user interface design, informed by expert feedback, indicates a positive impact on decision-making processes, demonstrating the prototype's ability to streamline vendor analysis and selection. The proposed tool significantly contributes to reducing the complexity and subjectivity of vendor selection, offering a more structured and data-driven approach.

**Keywords:** Software Ecosystem · Decision-making · User interface · Tool · Software Vendor Selection · Visualization

## 1 Introduction

The term Software Ecosystems (SECO) is used for groups of businesses, software service providers, customers and organisations, etc., which come together to develop software of mutual interest [15]. This approach emphasizes cooperative efforts on a common platform, aiming to tackle obstacles and achieve collective objectives [9, 1]. Rani et al. [10] categorised the entities participating in software ecosystems as: Company (outsourcing the software development), End User (generating demand for software), and Service provider/vendors (providing their services to the company) which are also called as collaborators. Choosing the most suitable vendor is challenging due to the availability of multiple vendors in a software ecosystem [9]. Typically, the decision-making task is carried out by

company personnel termed as decision-makers (DMs) [11, 10, 7]. The decision-making process becomes complex because decision-makers need to consider how each vendor fits into the company's strategic and architectural goals [10]. Making the wrong choice can result in serious consequences for the organization. Not only that, but the pressure intensifies further due to the ad-hoc and manual nature of today's decision-making processes [10], making them time-consuming and error-prone. Assisting decision-makers through these processes has been a large focus in the literature; however, limited research has been conducted on how information visualization and automated tools can support to increase the adaptability and usability of the underlying methods. Multiple studies [4, 2, 9, 14, 5] attempt to automate and aid decision-makers, however, they often overlook the importance of adaptability and user-friendliness in their approaches.

The clarity of requirements to be outsourced and the shared understanding among decision-makers and stakeholders is crucial due to the subjective nature of the task. DMs and stakeholders often oversee this, and transparency in communicating these requirements and vendor information is crucial in effective decision-making. Further, more than one decision-makers are involved in the

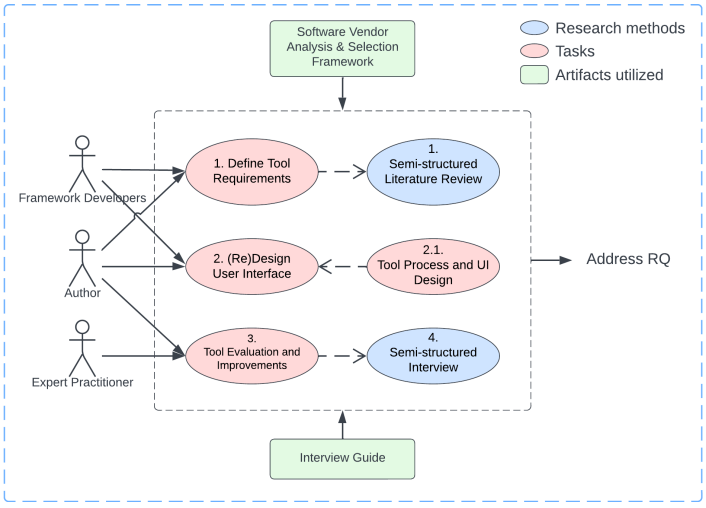


**Fig. 1.** Research methodology

decision-making process [13] generally, which further hinders the ease of reaching a common consensus of deciding on one vendor out of many. Considering these challenges of decision-makers in the software vendor selection process, integrating user interfaces and gathering information within one tool proves promising in helping decision-makers through these intricate processes. However, existing literature, instead of providing decision-makers with user interfaces, visual aids or automating the process, typically focuses solely on algorithms and methods.

Considering this research gap, the objective of this study is to expose a preliminary user interface of the tool to a practitioner. Hence, this paper evaluates the proposed user interface of the tool which supports decision-makers to gather and visualize vendor's data in one place in order to make informed decisions.

The foundation of the presented user interface is based on the framework by Rani et al. [9]. Hence, it is built in close collaboration with its authors to ensure its alignment with the framework. Further, it is tested and evaluated by a practitioner at the preliminary stage. Figure 1 shows the methodology consisting of various tasks performed to achieve the set objective, along with the research method employed to complete these tasks. Additionally, it shows the actors involved in completing tasks along with the required artefacts to conduct the study.

## 2 Tool Processes

After gaining a comprehensive understanding of the underlying framework and defining the tool's requirements including a strategy to consolidate essential data into a unified platform using minimalist design and sensible visualizations, we advanced to detail the tool's workflow. This effort laid the groundwork for how users would engage with the tool, steering the creation of a functional and intuitive interface.

Figures 2 and 3, illustrates the workflow of the tool through methodically designed steps and through methodically designed steps and thoughtful pause points. Aligned with the framework proposed by Rani et al. [9], it provides a systematic method from project initiation to RFP release, evaluation of VPs to vendor selection and recommendations. The process is methodically segmented into *Before RFP* and *After Receiving VPs* stages, as depicted in the respective figures, ensuring a streamlined path through the decision-making process.

**Before RFP:** The intended process flow is illustrated in Figure 2, and starts with initiating a new project, recognizing the need for outsourcing or integrating a new vendor. Stakeholders and decision-makers convene to define the project's specific requirements, leading to the drafting of the RFP document. This document outlines selection criteria for evaluating vendors, encompassing both primary and secondary factors for clear understanding among all parties. Subsequently, decision-makers assign weights to these criteria, indicating their relative importance. This crucial step, completed individually, precedes the use of an algorithm to check for inconsistencies or conflicts through the Inconsistency and Conflict Removal (ICR) method upon completion by all decision-makers. Detected issues prompt suggested resolutions, such as weight adjustments, discussions or other company procedures, ensuring consistency. This iterative resolution process ensures all conflicts are addressed before distributing the RFP to potential vendors for their proposals.

**After Receiving VPs:** The process flow of the tool after receiving VPs is illustrated in Figure 3. After the RFPs are released, vendors submit their proposals, detailing their solutions and alignment with the RFP requirements as a textual

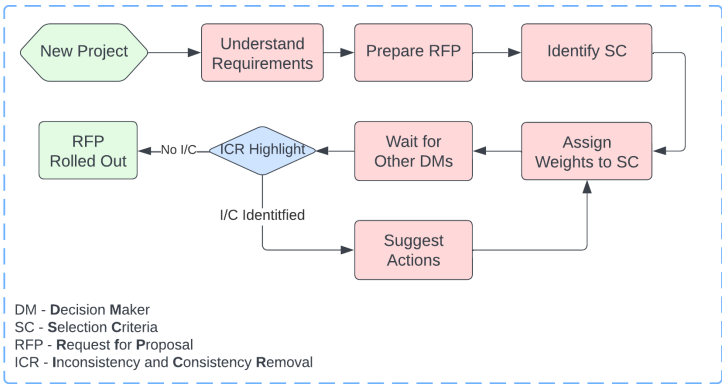


**Fig. 2.** Process flow of the tool for vendor analysis and selection (before RFP is rolled out)

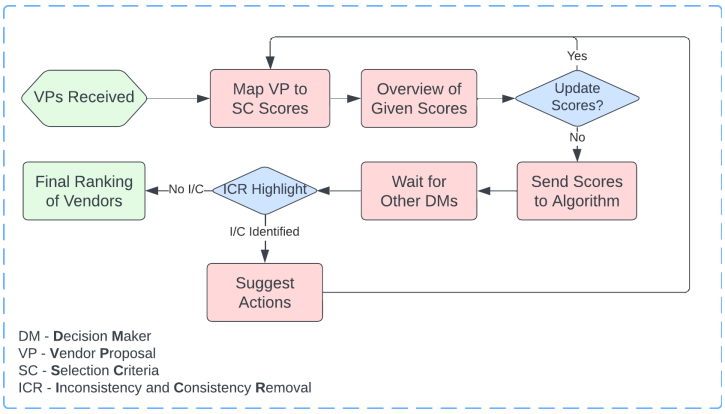


**Fig. 3.** Process flow of the tool for vendor analysis and selection (after VPs are received)

document. These VPs are then uploaded to the tool for decision-maker review, where they are evaluated against the selection criteria by assigning scores to each vendor.

Following the scoring, decision-makers receive visualizations summarizing vendor rankings and comparisons, helping to spot any potential scoring errors. They can adjust scores if needed before finalizing their evaluations for algorithmic analysis. The algorithm waits for all decision-makers to submit their scores, then checks for inconsistencies or conflicts in the scoring. If disagreements are identified, resolution steps are taken. Once resolved and no significant disagreements remain, the algorithm proposes the final vendor ranking for decision-makers and stakeholders consideration and action.

## 3 User Interface Evaluation and Improvement

This section presents some of the insights received from the practitioner (Table 1) so that the original voice of the practitioner can reach to the reader. This is divided into two parts: that is where the practitioner reinstates the need of the tool and the proposed user interface and secondly the additional functionality suggested to add on to the proposed user interface.

**Table 1.** Insights from practitioner

| **Theme** | **Practitioner (P)** |
|---|---|
| User interface evaluation | '...Sounds like a good idea to be honest. Very good idea. I don't know why we didn't think about an idea like this before. That's a very good idea, to be honest. I'm thinking on the possibilities of like, how can we use the same thing?' |
| | Depending on the size of the project and what kind of project it is. So 60 to 70 criteria and let's say average size of a project is having around 10 vendors. Like 60-70 criteria multiplied by ten and then comparing each vendor and each criteria with different thing it will be time consuming... |
| Suggested improvements | . . . We do phone, emails, everything, but if the system itself is having this option in the criteria, OK I'm checking this vendor in this section and I have a question for the vendor itself, so I'll write the question and send it and the system will maybe have the e-mail of the vendor so it will go to the vendor and come back here. So everything is transparent. . . . |
| | . . . .Yeah, because for like, transparency reason also ... because we call them which is bad... I should just write an e-mail with my question, it should go to the person without knowing my name or I know his name. He just replies me back. Because in our business, we know each other, we shouldn't know who is asking what. . . . |
| | . . . This will be helpful because we just write the page number and everything, and the person who wants to check it, they have to do it manually. . . . |

### 3.1 Tool Evaluation

The designed prototype integrates and enhances the framework by Rani et al. [9] for software vendor selection, streamlining project setup to final vendor ranking for decision-makers. It automates tasks and offers a semi-structured approach, centralizing information and calculations to overcome the manual and ad-hoc process issues identified by Rani et al. [9, 10] in previous studies, as well as by the practitioner (Table 1).

This tool not only simplifies the decision-making journey but also contributes to its overall success by reducing errors, enhancing efficiency, and ensuring a more systematic and data-driven approach which will be addressed next. The tool's development aligns with the research findings of Killen et al. [6], which underscore the value of visualization tools in empowering decision-makers to make more informed and successful choices [3, 6]. Consequently, it emerges as a crucial addition to the decision-making process. The tool integrate crucial decision-making data onto a single platform. It originally planned to conduct relative ranking of vendors based on the methodology of Rani et al. [9], which

employs AHP [12] as a baseline. However, the practitioner highlighted challenges with this method due to the large volume of VPs (often 10-30) and the extensive selection criteria list (around 60-70) typical in larger projects (Table 1). The necessity for this change caused a reevaluation of the process, ultimately resulting in a solution where decision-makers can evaluate a vendor against the criteria only, while having the option of viewing the pair-wise matrix as a secondary choice. The mapping of scoring will therefore be moved away from the decision-makers and onto the Application Layer and API. A screenshot from the vendor evaluation step can be seen in Figure 4. After all the vendors are

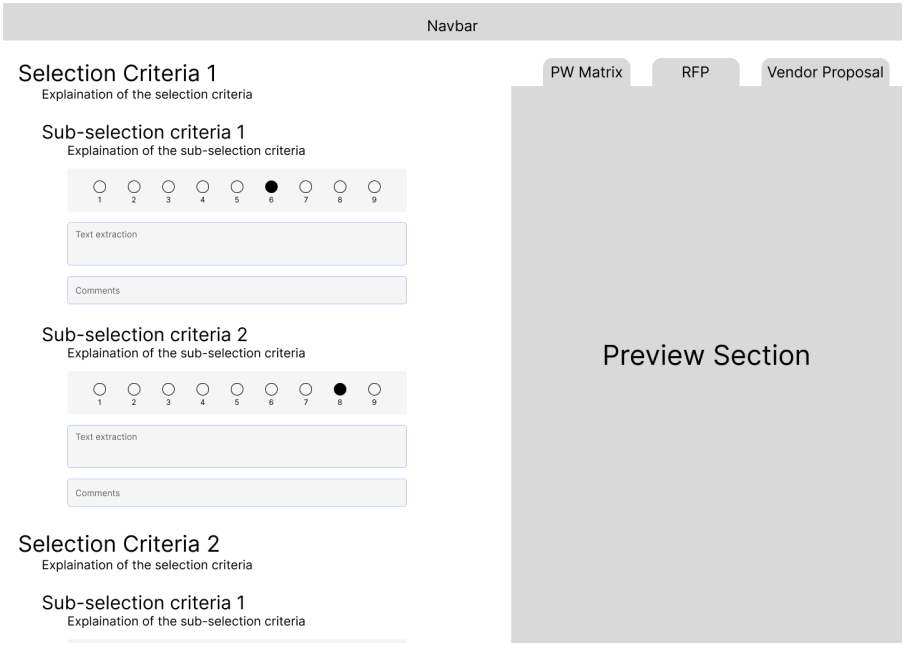


**Fig. 4.** Example of how to score vendors based on sub-criteria.

assessed against the selection criteria, a summary screen displays the vendors' scores for the decision-makers to review before submission. The authors of this study believe that employing visualizations in such cases can aid in vendor comparison, as also suggested by Alwi et al. [8]. Consequently, the tool includes a visualization screen after each vendor is evaluated to assist decision-makers in identifying any overlooked details or mistakes when comparing vendors to each other. An example of this visualization page can be seen in Figure 5. According to the insights received, until now, decision-makers heavily rely on manual and ad-hoc processes for software vendor selection, which involved complicated tasks like writing down page numbers and manually checking each entry. This process is time-consuming and error-prone, making decision-makers wish for a

streamlined process from project setup till final vendor ranking. Thus, the tool presented in this study will benefit decision-makers in automating the tasks, and centralize information to overcome these issues. Adding to the processes being labor-intensive, the practitioner also emphasized that in practice, they lack a systematic approach, leading to inefficiencies and potential oversights in the decision-making process. Thus, one of the aids of the proposed tool will be to provide all decision-makers with a similar semi-systematic and data-driven evaluation of software vendors, in the hopes of enhancing decision-making efficiency by reducing errors. Additionally, the practitioner highlighted the significance of consolidating all decision-making data onto a single platform, emphasizing how the proposed tool can simplify the decision-making journey and contribute to its success. Lastly, the practitioner also highlighted that prior practices did not effectively leverage visualizations in the decision-making process. Since the proposed tool carefully integrates visualizations into parts of the decision-making process, this further supports decision-makers in taking more informed decisions, along with easily comparing vendor data at one place.

### 3.2 Suggestions for Improvement

The practitioner further emphasized that the community is small where many of them know each other and that therefore, providing the tool with an anonymous communication feature will enhance transparency and eliminate bias (Table 1). In a nutshell, following suggestions to improve the user interface further are sug-

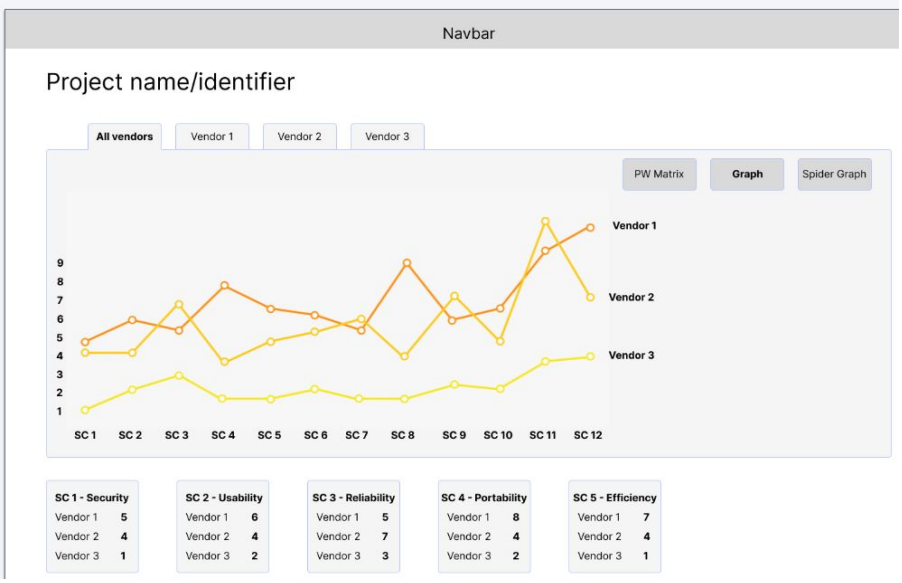


**Fig. 5.** Example of vendor summary view for a decision maker

gested by the practitioner:
**Direct, Anonymous Communication with Vendors:** Allowing decision-makers to anonymously get in touch with the vendors to resolve any questions or uncertainties that may arise while conducting the vendor analysis is asked for by the practitioner. This could include functionalities for sending emails or initiating chats. This feature is asked for so that the decision-makers become more informed before making a decision, reducing or removing bias through anonymous contact, and ensuring a transparent and efficient communication process.
**Flexible Scoring System:** Furthermore, implementing a flexible scoring system is highlighted as a wanted feature. This is partially included already, but emphasis is put on including and further developing this process. Allowing decision-makers to score both overarching selection criteria as well as granular sub-criteria and questions accommodate the diverse and unique requirements of different projects, allowing for a tailored and nuanced vendor evaluation process.
**Text Extraction and Selection Criteria Mapping:** Lastly, developing capabilities to automatically extract and map text from RFPs and VPs to relevant selection criteria scores are asked for by the practitioner. This feature aims to streamline the evaluation process by reducing manual data entry and enhancing the accuracy and efficiency of mapping vendors' offerings to decision-makers' criteria.

## 4 Conclusion, Limitation and Future Work

This study explored the critical area of software vendor selection in software ecosystems, a domain where decision-making is paramount yet challenging due to the complexity and variety of factors involved. Recognizing the necessity for a robust tool with a reliable user interface for the decision-making process, we proposed user interface sketches and workflow of the potential tool that leverages the power of visualizations to enhance decision-making capabilities. By integrating a user-friendly interface with the underlying methodology of the Rani et al. [9] framework and the established need of such kind of tool interface, this study paves new paths for investigating the role of user interfaces and visualization in decision-making processes for researchers. It highlights the importance of integrating practical tools with theoretical frameworks to enhance the utility and applicability of research findings. However, the reliance on a limited body of literature and feedback from a single practitioner may affect the generalizability of the findings presented in this work. Hence, this work is at a preliminary stage. In the future, we intend to develop the initial functional prototype and expose it to a variety of practitioners. Additionally, expanding the scope to include more diverse methodologies and integrating advancements in visualization and decision-making theories could further refine and enhance the tool's capabilities.

## Bibliography


[1] Simone Amorim et al. "Tailoring the NFR Framework for Measuring Software Ecosystems Health". In: Jan. 2018. DOI: 10.17771/PUCRio.wer.inf2018-14.

[2] Özge Deretarla, Babek Erdebilli, and Mete Gündoğan. "An integrated Analytic Hierarchy Process and Complex Proportional Assessment for vendor selection in supply chain management". In: *Decision Analytics Journal* 6 (2023), p. 100155. ISSN: 2772-6622. DOI: https://doi.org/10.1016/j.dajour.2022.100155. URL: https://www.sciencedirect.com/science/article/pii/S2772662222000868.

[3] Karin Eberhard. "The effects of visualization on judgment and decision-making: a systematic literature review". In: *Management Review Quarterly* 73.1 (Feb. 2023), pp. 167–214.

[4] Cevriye Gencer and Didem Gürpinar. "Analytic network process in supplier selection: A case study in an electronic firm". In: *Applied Mathematical Modelling* 31.11 (2007), pp. 2475–2486. ISSN: 0307-904X. DOI: https://doi.org/10.1016/j.apm.2006.10.002. URL: https://www.sciencedirect.com/science/article/pii/S0307904X06002368.

[5] Ozcan Kilincci and Suzan Aslı Onal. "Fuzzy AHP approach for supplier selection in a washing machine company". In: *Expert Systems with Applications* 38.8 (2011), pp. 9656–9664. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2011.01.159. URL: https://www.sciencedirect.com/science/article/pii/S0957417411001928.

[6] Catherine P. Killen, Joana Geraldi, and Alexander Kock. "The role of decision makers' use of visualizations in project portfolio decision making". In: *International Journal of Project Management* 38.5 (2020), pp. 267–277. ISSN: 0263-7863. DOI: https://doi.org/10.1016/j.ijproman.2020.04.002. URL: https://www.sciencedirect.com/science/article/pii/S0263786320300260.

[7] Hideki Kobayashi and Hiroshi Osada. "Strengthening of collaboration between IT vendors and their customer through proposal-based sales". In: *2011 IEEE International Conference on Quality and Reliability*. Sept. 2011, pp. 556–560. DOI: 10.1109/ICQR.2011.6031601.

[8] Nik Nur Ayuni Nik Alwi et al. "Data Visualization of Supplier Selection Using Business Intelligence Dashboard". In: *Advances in Visual Informatics*. Ed. by Halimah Badioze Zaman et al. Cham: Springer International Publishing, 2019, pp. 71–81. ISBN: 978-3-030-34032-2.

[9] Anshul Rani, Deepti Mishra, and Aida Omerovic. "A framework for software vendor selection by applying Inconsistency and Conflict Removal (ICR) method". In: *International Journal of System Assurance Engineering and Management* (Nov. 2023). ISSN: 0976-4348. DOI: 10.1007/s13198-023-02190-x. URL: https://doi.org/10.1007/s13198-023-02190-x.

[10] Anshul Rani, Deepti Mishra, and Aida Omerovic. "Exploring and Extending Research in Multi-vendor Software Ecosystem". In: *Innovations in Smart Cities Applications Volume 5*. Ed. by Mohamed Ben Ahmed

et al. Cham: Springer International Publishing, 2022, pp. 379–391. ISBN: 978-3-030-94191-8.

[11] Anshul Rani, Deepti Mishra, and Aida Omerovic. "Multi-vendor Software Ecosystem: Challenges from Company' Perspective". In: *Information Systems and Technologies*. Ed. by Alvaro Rocha et al. Cham: Springer International Publishing, 2022, pp. 382–393. ISBN: 978-3-031-04829-6.

[12] R.W. Saaty. "The analytic hierarchy process—what it is and how it is used". In: *Mathematical Modelling* 9.3 (1987), pp. 161–176. ISSN: 0270-0255. DOI: https://doi.org/10.1016/0270-0255(87)90473-8. URL: https://www.sciencedirect.com/science/article/pii/0270025587904738.

[13] Thomas L. Saaty and Luis G. Vargas. "Uncertainty and rank order in the analytic hierarchy process". In: *European Journal of Operational Research* 32.1 (1987), pp. 107–117. ISSN: 0377-2217. DOI: https://doi.org/10.1016/0377-2217(87)90275-X. URL: https://www.sciencedirect.com/science/article/pii/037722178790275X.

[14] Giustina Secundo et al. "Supporting decision-making in service supplier selection using a hybrid fuzzy extended AHP approach". In: *Business Process Management Journal* 23.1 (Feb. 2017), pp. 196–222. DOI: 10.1108/bpmj-01-2016-0013. URL: http://dx.doi.org/10.1108/bpmj-01-2016-0013.

[15] Eric Yu and Stephanie Deng. "Understanding software ecosystems: A strategic modeling approach". In: *Iwseco-2011 software ecosystems 2011. proceedings of the third international workshop on software ecosystems. brussels, belgium*. 2011, pp. 65–76.

# Appendix B

# Previous Work

Addressing previous work conducted before the thesis period in other courses is deemed necessary. Three preliminary studies were conducted to build a foundation for the thesis.

A preliminary literature review was conducted in the *Research Project Planning* course[1] on the topics of Software Ecosystems, identification of selection criteria and vendor selection methods to identify gaps and limitations.

During the Research and Development course *Advanced Project Work*[2], a semi-structured literature review was conducted on the limitations in existing frameworks and tools, the role of visualizations in decision-making and the cognitive challenges in visualization. The findings served key in conducting the first iteration of action research which concerned the initial requirement definition, tool process flow and low-fidelity user-interface designs. Finally, the evaluation with the single practitioner was also conducted during this iteration and course.

Selecting the articles for the systematic litearture review conducted on non-functional requirements as selection criteria was executed in the *Advanced Topics in Software and Systems Engineering*[3] course. Initial results such as demographics data and year-wise distribution were also presented there, while the in-detail analysis and interpretations were made during the thesis period.

These parts are re-written and interpreted over again to fit the new format and context of the thesis, however, it is worth having in mind as these will not be stated more times throughout the thesis work as they disrupt the flow and readability of the thesis.

[1] Course info is available at: `https://www.ntnu.edu/studies/courses/MACS4000#tab=omEmnet`

[2] Course info is available at: `https://www.ntnu.edu/studies/courses/IMT4894#tab=omEmnet`

[3] Course info is available at: `https://www.ntnu.edu/studies/courses/IMT4889#tab=omEmnet`

# Appendix C

# Interview Guide

## C.1 First Iteration

### C.1.1 Preparation and Other Details

Invitation for the interview was sent out by my supervisor to all participants.

The interview was held remotely through MS Teams as the participants were located in different places around the world at the time of the interview.

The interview had a semi-structured approach, having a predefined set of questions as well as allowing the interviewers to adapt and ask additional questions where sensible. In addition to having a predefined set of questions, the prototype was presented to the decision-maker allowing them to come up with feedback whenever they could think of anything. The questions that were used for this interview are presented below in no particular order:

### C.1.2 Questions Guide

1. Could you briefly go through the process of how you do the vendor selection process?
2. When evaluating vendor proposals, how do decision-makers assess vendor proposals in terms of scoring?
    a. Do you want to assign scores to criteria of vendors individually or do you want to score vendors in relation to each other?
3. Do you recommend showing the weights of each criterion to the decision-maker?
4. How much do decision-makers utilize the pair-wise matrix and in what cases?
5. How do decision-makers prefer to map the selection criteria to the RFP and vendor proposal to the selection criteria?
6. Does decision-makers wish to have the SC and their explanation or rather answer questions that are mapped to these criteria (what level of hierarchy)?

   a. Selection Criteria
   b. Sub-Selection Criteria
   c. Questions

7. What do you think of having the vendor proposals and request for proposals etc., within the tool and allow decision-makers to do text extractions from these and map those to the selection criteria?
8. How do you prefer highlighting conflicts? Should the decision makers get to see each others scores for instance, or should they just get to know that there has been a conflict for that selection criteria and that action should be taken accordingly?

*Presenting the prototype and steps for open feedback from the decision-maker*

## C.2 Second Iteration

1. **Introductory tool evaluation:**
   Are you using any tools, or how do you do this process (decision-making/software vendor selection) now?
   (a) If yes, which ones and what are some of their strengths and weaknesses?
   (b) If no, how do you do this process now?
2. **Selection process and criteria:**
   How does the selection criteria process (specifically NFRs) look in your vendor selection process, either formally or informally?
   (a) What are the main criteria for this selection process (if you are able to identify any criteria that most of the time are there)?
3. **User roles and responsibilities:**
   How do you consider transparency and bias during the decision-making process?
   (a) The tool tries to provide transparency and remove bias through measures such as only providing higher authorities with the final vendor ranking and decision-makers not knowing who else is decision-makers for that project, nor being able to see their scoring, inconsistencies, and conflicts.
      (i) Could you provide your views about such measures for the process? And are there other considerations you would have liked to see in such a tool?
4. **Tool effectiveness and Usability (usefulness and usability):**
   In what ways do you think this tool can impact your vendor selection process?
   (a) Are there specific tasks that the tool has made easier?
   (b) What was the interesting/most useful parts of the tool? What would you like to add in addition?
5. In your opinion, how intuitive do you find the user interface and overall user experience of the tool?
   (a) Could you provide some specific examples, if you have any?
6. **Features and Functionality:**
   Are there any features in the tool that you find particularly valuable? Why?
7. What features do you think are lacking or could be improved in the tool?
   (a) How would these improvements help you?
8. **Overall impact:**
   What do you think about the use of such a tool in your organization/decision-

making process?

   (a) In what way will it support you during this process?

      (i) How effective do you find the tool to be in helping make these decisions?

9. Would you like to have some tool like this included in the decision-making process in your organization?

   (a) If yes, why? If no, why not?
   (b) Do you think it could be feasible/useful in its current state? Optionally, what should be added to make it feasible for your organization?

10. Overall, how satisfied are you with the tool in its current form?
11. How does this tool compare to any other tools or methods you have used in the past for vendor selection?
12. Are there any features that you believe should be deemed essential in tools designed to assist decision-makers in selecting software vendors, whether they are present in the proposed tool, in other similar tools, or they come to your mind now?

Before rounding off, do you have any additional feedback or suggestions that you would like to share about the tool or in general?

# Appendix D

# Initial Prototype Requirements

These main feature requirements, in no particular order, are:

*The tool should*

- show vendor proposals
- show request for proposal
- show pair-wise matrix
- show selection criteria and explanations
- let decision-makers assign scores to selection criteria either as
    - main criteria
    - sub-criteria
    - questions
- let decision-makers/stakeholders assign new criteria to the vendor selection process
- let decision-makers/stakeholders assign weights to the criteria
- highlight conflicts and inconsistencies in weights for the criteria
- highlight conflicts and inconsistencies in weights for vendor capabilities
- suggest actions to take based on conflict
- summarize vendor rankings and visualize data
- show final vendor rank

# Appendix E

# Systematic Literature Review Inclusion and Exclusion Protocol

## E.1 Selection Protocol

Below is the selection protocol for the SLR. This protocol was used to evaluate papers and proved especially important in Step 3 (section E.2.3) of the study selection process as at this stage two reviewers had to go through the papers individually.

### E.1.1 Inclusion and exclusion criteria

The following inclusion and exclusion criteria will be used for data extraction of the literature found by search terms:

**Inclusion criteria**

- Studies that describe software ecosystems/software outsourcing relationships
- Studies that describe decision-making/vendor selection in a software outsourcing context
- Studies that describe non-functional requirements/quality attributes in a software ecosystem context
- Studies that describe selection criteria in software outsourcing projects
- Studies that describe modeling- and architecture techniques in a software ecosystem context

**Exclusion criteria**

- Incomplete full-text availability
- Papers that are of type conference reviews, books, keynotes, presentations
- Other systematic literature reviews
- Studies published before 2013
- Publication language is not English
- Studies that do not consider the software domain

## E.2 Selecting primary sources

### E.2.1 Step 1 - Collection of all papers

The initial phase involved acquiring the papers for examination. After acquiring four papers that were used as seed studies the initial snowballing process began to gain an overview of the software ecosystem domain, encompassing decision-making processes and criteria. Once this preliminary exploration concluded, a foundational understanding of decision-making in software ecosystems was established, setting the stage for defining queries in preparation for the SLR. This step involved the crucial task of keyword and search term definition, considering the various ways terms can be expressed in the software engineering domain, including alternative spellings and synonyms. Striking the right balance between depth and breadth was challenging, aiming to capture relevant papers without overlooking them. After refining the search query through trial and error, the first database search was conducted in Scopus, with subsequent adjustments for compatibility with other databases.

The total number of papers retrieved from all databases was initially very high, ending at 45510 papers. Because of this, we had to scope down the study to select a maximum of the first 500 papers from each database, sorted by relevance. This resulted in choosing 500 papers from both ACM Digital Library and IEEE Explore, 382 papers from Scopus, and all four results from SpringerLink were excluded based on exclusion criteria. The distribution can be seen in Figure **??**.

Ensuring the exclusion of duplicate publications is crucial in a systematic review synthesis to prevent potential bias resulting from the repetitive presentation of the same data [67]. Following this, the included 1382 papers underwent a thorough removal of duplicates across databases, resulting in a total of 1305 unique papers. As we were only able to apply the wished-for filters to some of the databases, 30 papers had to be removed due to paper-type filtration (for instance conference reviews) resulting in us having 1285 papers left for inclusion in the subsequent step.

### E.2.2 Step 2 - Removal based on titles and protocol

Given the inclusion of 1285 papers and time constraints for the review, the subsequent step involved the classification of papers into relevant and irrelevant categories based on their titles. This classification was necessary as many results were not directly related to the search field. While some filters could have been applied during the database searches to streamline this process, it was not universally feasible in all databases. Therefore, the exclusion work based on titles was conducted alongside the categorization of relevance. Papers were labeled as either satisfying the protocol and scope (yes) or not (no). Papers with ambiguous relevance were carried forward to the next step. This stage resulted in 79 papers advancing to Step 3.

### E.2.3 Step 3 - Removal based on abstracts and conclusions

The subsequent phase involved a thorough examination of abstracts and conclusions to categorize papers based on adherence to the protocol. To mitigate bias, two reviewers conducted this evaluation in separation using the research protocol. After the review was conducted by each of the reviewers, they got together and discussed their results. As shown by the Kappa Cohen Statistic evaluation (see Section E.3) the reviewers showed *substantial agreement*. After discussion among the reviewers, all the papers where the reviewers had conflicting interests in inclusion were included in the next step. Following this stage, 39 papers were selected for further review.

### E.2.4 Step 4 - Full-text evaluation

In the second to last step of the SLR, all 39 papers' full texts from Step 3 were extracted and comprehensively reviewed against the protocol. At this step another filtration of papers was done where an additional 8 papers were removed based on for instance relevance and availability, resulting in a final set of 31 papers. Furthermore, data was extracted from these papers to answer the research questions and to be able to further evaluate the information of the papers for the MSc thesis.

## E.3 Cohen Kappa Statistic

Cohen Kappa Statistics were used to measure the level of agreement between the two researchers during Step 3. It helps quantify the extent to which the assessments align, accounting for the possibility of agreement occurring by chance. Utilizing this statistic enhances the reliability of the review process by providing a numerical indicator of the consistency in the evaluations, thereby contributing to the overall credibility of the SLR findings.

The calculation of the level of agreement was done according to the Cohen Kappa Statistic formula [68], giving us a k-value of 0.70 which indicates *Substantial Agreement*. While the k-value ideally should have been as close to 1.0 as possible, we are happy with a value of 0.70. We believe that the offset is mainly due to the difference in both experiences within the field of software ecosystems, as well as experience in conducting SLRs.

| Rater 1 / Rater 2 | Yes | No | |
|---|---|---|---|
| Yes | 28 | 1 | 29 |
| No | 11 | 39 | 50 |
| | 39 | 40 | |

**Figure E.1:** Cohen Kappa

# Appendix F

# NFR Mapping to Our Context

Thus far, emphasis has been placed on scrutinizing the domains and motivations of the studies under review. In this section, a thorough examination of the findings relevant to each requirement will ensue. This entails providing a comprehensive overview and discussion of how these requirements are described across the various papers and the ISO/IEC 25010:2023 standard. Additionally, an evaluation of the appropriateness of these definitions for selecting software vendors within software ecosystems will be conducted. Where necessary, any gaps will be identified, and strategies for crafting independent interpretations of the requirements definitions will be outlined. Ultimately, the goal is to consolidate these insights to propose a set of requirements, definitions, and sub-criteria. The objective of these is to establish a robust foundation for stakeholders to provide decision-makers with during the vendor selection process.

***Availability***

The first requirement of interest is *Availability*. Three of the papers that address availability also have a definition for the requirement. Furthermore, availability is also defined in the ISO/IEC 25010:2023 standard.

Belinda et al. [29] defines availability as:

*Availability refers to the degree to which a software product is operational and easily accessible when needed for usage.*

Kumar et al. [71] defines availability as:

*Availability represents the percentage of time a cloud service is up without interruption.*

Sharma et al. [55] suggest that availability is:

*It is the standard to which software or a component is available for use, access or operation.*

While Lytra et al. [9] does not suggest any definition or explanation of availability, leaving it to the decision makers' subjectivity.

The ISO/IEC 25010:2023 standard address availability as a sub-attribute of reliability and define it as:

*Capability of a product to be operational and accessible when required for use.*

The term *availability* may have different meanings depending on the field it is applied to and the eye who sees it, as showcased in the literature findings above. We believe that the papers fail to capture the entirety of such a definition in software vendor selection context. Thus, we believe it is sensible to develop our own clear and specific definition and recommendation to ensure a consistent and unambiguous understanding of what *availability* entails in the context of selecting software vendors within a software ecosystem. Our suggestion has its foundational understandings and interpretations from the literature findings, and standards such as ISO/IEC 25010:2023, as well as our take on it for the software ecosystem context.

We believe that for the software ecosystem context, availability can be viewed both in the context of the software the vendor provides, but also as the availability of the vendor itself.

Based on the ISO/IEC 25010:2023 standard, Belinda et al. [29], Sharma et al. [55] and our interpretations, we suggest an overall definition of *availability* to be:

*Software availability in an ecosystem denotes the product's and vendor's ability to ensure continuous access and functionality. It combines the software's reliability with vendor support to minimize downtime, ensuring the product is operational and accessible as needed, according to defined performance standards and agreements.*

Sub attributes:

| | |
|---|---|
| **Availability** | System Uptime Guarantees |
| | Support Response Time |
| | Update Frequency |

As mentioned earlier, the existing definitions of the criterion do not fully encompass the broader scope of interpretations applicable to software vendor selection. It is essential to consider this criterion in the context of both vendor and product availability to ensure that decision-makers grasp the complete landscape, rather than solely concentrating on product availability, as is often the case. Both the product and the vendors themselves play vital roles in the success of ecosystems. Thus, by adopting this approach, decision-makers are likely to enhance their ability to select vendors that adhere to the necessary reliability and operational standards crucial for successful software ecosystems.

***Scalability***

The second requirement of interest is *Scalability*. None of the included papers addressed a definition of the scalability criteria or its sub-criteria, leaving the definition from the literature empty.

The ISO/IEC 25010:2023 standard address scalability as a sub-criteria of Flexibility, but define scalability as:

*Capability of a product to handle growing or shrinking workloads or to adapt its capacity to handle variability.*

We believe that when considering scalability as a selection criterion in the software vendor selection context, it is essential to assess both the technical capabilities of the software product and the vendor's ability to support the organization's growth.

Based on the ISO/IEC 25010:2023 standard and our interpretations, we suggest an overall definition of *scalability* to be:

*Scalability in a software ecosystem highlights the product's and vendor's capability to support growth, emphasizing the ability to adapt to increasing demands and business needs. It encompasses the software's customization flexibility and global expansion capabilities, ensuring that solutions not only evolve with but also support performance consistency across different regions.*

| | |
|---|---|
| **Scalability** | Complexity |
| | Extensibility |
| | Interoperability |
| | Performance |

Given the absence of scalability definitions beyond those outlined in the ISO/IEC 25010:2023 standard, and particularly in the context of software vendor selection, there arises a necessity for a new definition. While the standard's definition adequately addresses product scalability aspects, it overlooks vendor capabilities considerations. The proposed redefinition encompasses both product and vendor capabilities pertaining to scalability, specifically within the domain of software vendor selection. This approach aims to enhance decision-makers' capacity to make well-informed and ultimately successful decisions when considering scalability.

***Portability***

The third requirement of interest is *Portability*. Despite five of the papers addressing portability, only three of which provide a definition of the criteria. Furthermore, the criterion is also defined in the ISO/IEC 25010:2023 standard.

Jansen et al. [72] vaguely defines portability as:

*Portability describes the platforms that are required to install the main platform.*

Belinda et al. [29] defines portability as:

*The measure of the ease of transferring software from one computing environment to the other.*

Amorim et al. [4] defines portability as:

*Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware or software environment to another.*

The ISO/IEC 25010:2023 standard have renamed portability to *flexibility*, and define it accordingly:

*Capability of a product to be adapted to changes in its requirements, contexts of use, or system environment.*

The current definitions derived from literature and the ISO/IEC 25010:2023 standard provide valuable insights into the portability requirement. Nevertheless, we contend that by integrating the strengths of each definition, a more robust and comprehensive definition can be formulated for the benefit of decision-makers in software vendor selection.

Based on the ISO/IEC 25010:2023 standard, Jansen et al. [72], Belinda et al. [29], Amorim et al. [4] and our interpretations, we suggest an overall definition of *portability* to be:

*Portability in a software ecosystem encapsulates the software's flexibility to adapt and function across diverse computing environments without extensive modification. It merges the principles of platform independence, data migration efficiency, and interoperability, ensuring seamless operation and integration, regardless of the underlying technology.*

Sub attributes:

| **Portability** | Adaptability |
|---|---|

While the existing definitions found in literature and the ISO/IEC 25010:2023 standard offer valuable insights into the portability requirement, we propose that by combining elements from these diverse definitions and consolidating their aspects, decision-makers can develop a more comprehensive understanding of portability, particularly within the context of software vendor selection. Moreover, in addition to clearly and concisely summarizing the findings, this definition explicitly addresses principles such as platform independence, data migration efficiency, and interoperability, all of which are crucial factors in evaluating portability. Consequently, through this refined definition, decision-makers are better equipped to assess both vendors and their products' portability.

***Performance***

The fourth requirement of interest is *Performance*. While five of the papers under review addressed performance, only two of which, in addition to the ISO/IEC 25010:2023 standard, provided a definition of the criterion.

Belinda et al. [29] defines performance as:

*Performance refers to the total effectiveness of a software product.*

Ameller et al. [73] defines performance based on the ISO/IEC 25000 definition, making performance efficiency:

*Performance relative to the amount of resources used under stated conditions.*

The ISO/IEC 25010:2023 standard address performance "performance efficiency" and describes it as:

> *Capability of a product to perform its functions within specified time and throughput parameters and be efficient in the use of resources under specific conditions.*

The definitions of performance derived from literature and the ISO/IEC 25010:2023 standard offer valuable insights. However, we advocate for a refined interpretation to include vendor performance, as the existing definitions overlook this aspect.

Based on the ISO/IEC 25010:2023 standard, Belinda et al. [29], Ameller et al. [73] and our interpretations, we suggest an overall definition of *performance* to be:

*Performance in the context of software ecosystems involve evaluating the software's efficiency in resource utilization and operational effectiveness, including scalability, response times, and reliability. Concurrently, it requires assessing the vendor's ability to meet service level agreements, innovate, and adapt to technological advancements, ensuring they provide robust support and continuous improvement.*

Sub attributes:

| | |
|---|---|
| **Performance** | Capacity |
| | Resource Utilization |
| | Timing Behaviour |

While existing definitions found in literature and the ISO/IEC 25010:2023 standard provide valuable insights into the concept of software product performance, we advocate for a refined interpretation. By combining elements from these varied definitions and also addressing performance in vendor context, the aim is to offer decision-makers a more nuanced understanding of performance, especially concerning software vendor selection. Additionally, the proposed definition articulates key principles such as efficiency in resource utilization, operational effectiveness, scalability, response times, and reliability, which are vital in assessing software products and vendors. Consequently, this redefined perspective equips decision-makers with a more comprehensive requirement understanding to evaluate vendors and their products' performance effectively, addressing the evolving needs and complexities of software ecosystems.

***Maintainability***

The fifth requirement of interest is *Maintainability*. Maintainability is addressed in seven of the papers, but only three papers in addition to the ISO/IEC 25010:2023 standard, provided a definition of the criterion.

Amorim et al. [4] defines maintainability as:

> *Degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.*

Amorim et al. [74] does not explicitly define maintainability, but they talk about it in a way when looking at maintainability and productivity against each other, in which we may extract the main ideas into a definition:

*Efforts of new members to refactor and improve the code made the system usable. In its turn, the ecosystem effectively started again to deliver new technologies, processes, and ideas to its members, influencing the productivity.*

Belinda et al. [29] defines maintainability as:

*The ease with which software can be modified to correct faults or improve performance.*

Belinda et al. [29] also defines maintainability's sub-criteria *Flexibility* as:

*Flexibility is the ability of software to adapt to possible future changes in its requirements.*

The ISO/IEC 25010:2023 standard address maintainability as:

*Capability of a product to be modified by the intended maintainers with effectiveness and efficiency.*

Although the definitions offered by the literature and the ISO/IEC 25010:2023 standard establish a solid foundation for maintainability, we propose that consolidating the strengths of these definitions into a singular definition tailored for the software vendor selection context would be beneficial for decision-makers.

Based on the ISO/IEC 25010:2023 standard, Amorim et al. [4, 74], Belinda et al. [29] and our interpretations, we suggest an overall definition of *maintainability* to be:

*Maintainability in a software ecosystem reflects the software's and vendor's capability to efficiently and effectively modify the product to correct faults, enhance performance, or adapt to changing requirements.*

Sub attributes:

| | |
|---|---|
| **Maintainability** | Flexibility |
| | Extensibility |
| | Supportability |

While the current definitions offer a robust foundation for maintainability, we believe it is beneficial to customize the definition for the software vendor selection context. Additionally, by increasing support for decision-makers with more detailed examples of maintenance tasks — such as correcting faults, improving performance, and adapting to changing requirements — we anticipate that decision-makers will find it easier to assess vendor and product maintainability based on the Request for Proposal (RFP) and vendor proposals.

***Usability***

The sixth requirement of interest is *Usability*. Usability is addressed in eight of the papers under review, but only provided with a definition in four of them, in addition to the ISO/IEC 25010:2023 standard.

Kocak et al. [75] defines usability as:

*A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.*

Amorim et al. [74] defines usability as:

*Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.*

Belinda et al. [29] discusses usability in this context:

*Krita has a flexible interface to users configure as they want and provides a tutorial to users coming from other applications. A more usable interface facilitates the engagement of new users and allows the ecosystem to keep its capacity of growing.*

Chazette et al. [76] defines usability as:

*The ease of use and learnability of software by customers.*

The ISO/IEC 25010:2023 standard have renamed usability to *interaction capability* and define it as:

*Capability of a product to be interacted with by specified users to exchange information between a user and a system via the user interface to complete the intended task.*

The definitions of usability found in existing literature and the ISO/IEC 25010:2023 standard offer valuable insights. However, we propose that by combining the definitions from both sources, decision-makers can benefit from a more refined and clear definition, particularly in the context of software vendor selection.

Based on the ISO/IEC 25010:2023 standard, the four papers' definitions [29, 74–76], and our interpretations, we suggest an overall definition of *usability* to be:

*Usability in a software ecosystem denotes the software's capacity to provide an intuitive, learnable, and efficient interface that enables users to achieve their objectives with satisfaction in a specified context of use. It encompasses the system's adaptability to user needs, facilitating engagement and reducing error rates by guiding users towards correct actions and minimizing misunderstandings.*

| | |
|---|---|
| **Maintainability** | Operability (2x papers) |
| | Understandability (2x papers) |
| | Learnability |
| | Attractiveness |
| | Accessibility |
| | Appropriateness Recognisability |
| | Learnability |

Sub attributes:

The definitions of usability in existing literature and the ISO/IEC 25010:2023 standard provide valuable insights. However, we suggest merging aspects from these definitions to enhance decision-makers' understanding, especially regarding software vendor selection. This refined definition summarizes findings clearly and also explicitly covers fundamental principles, which are essential for evaluating usability. As a result, decision-makers can better assess if vendors' meet the criteria of usability for their products.

***Security***

The seventh requirement of interest is *Security*. While it is the requirement which is addressed in the largest number of papers, featuring ten papers, only two papers provide a definition of the criterion. Furthermore, the ISO/IEC 25010:2023 standard also provide a definition.

Dayanandan et al. [77] defines security based on their version of the ISO 25010 standard:

*The degree to which a product or system protects information and data so that persons or other products or systems have the degree of data appropriate to their types and levels of authorization.*

Amorim et al. [4] defines security as:

*Degree to which a product or system protects information and data so that persons or other products or systems have the data access appropriate to their types and levels of authorization.*

The ISO/IEC 25010:2023 standard address security as:

*Capability of a product to protect information and data so that persons or other products have the degree of data access appropriate to their types and levels of authorization, and to defend against attack patterns by malicious actors.*

The definitions sourced from the literature are primarily rooted in the previous iteration of the ISO 25010. We perceive these definitions, along with those from the ISO/IEC 25010:2023, as closely aligned with our requirements. Nonetheless, we recommend incorporating the vendor's capacity to secure information and data, particularly concerning information reserved for ecosystem participants. Additionally, we propose a slight adjustment to the flow of the definitions.

Based on the ISO/IEC 25010:2023 standard, Dayanandan et al. [77], Amorim et al. [4] and our interpretations, we suggest an overall definition of *security* to be:

*Security in a software ecosystem is defined by the software's and vendor's ability to safeguard information and data, ensuring access is strictly provided according to the users' authorization levels and effectively protecting against unauthorized access and malicious threats.*

Sub attributes:

| | |
|---|---|
| **Security** | Confidentiality (2x papers) |
| | Integrity (3x papers) |
| | Non-Repudiation (2x papers) |
| | Accountability |
| | Authenticity (2x papers) |
| | Compliance |
| | Access control |
| | Encryption |
| | General |
| | Authentication |

While the definitions of security in existing literature and the ISO/IEC 25010:2023 provide valuable insights, a slight modification to include the vendor's capability to secure information and data is suggested. By doing so, decision-makers will need to carefully examine the vendor more thoroughly, not just the product they offer. This could involve steering clear of vendors with a history of mishandling critical organizational data, which could potentially harm the ecosystem either as a whole or in terms of competitiveness.

***Reliability***

The eight and last requirement of interest from the selection of papers is *Reliability*. Reliability is highlighted in nine of the papers and features definitions in five of which, in addition to the definition from ISO/IEC 25010:2023.

Lytra et al. [9] defines reliability as:

*Degree to which a system, product or component performs specified functions under specified conditions for a specified period.*

Amorim et al. [74] defines reliability as:

*Reliability refers to the probability of software operating in a given environment within a specified period to perform well without encountering a breakdown.*

Ameller et al. [73] defines reliability as:

*A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.*

Kumar et al. [71] defines reliability as:

*It refers to the probability of successful response of the cloud service for a given period of time and condition.*

Kocak et al. [75] defines reliability based on the IEEE definition and defines it as:

*Software reliability is the ability of a system or component to perform its required functions under stated conditions for a specified period of time. Software becomes unreliable due to software failures which occur as a result of software errors.*

The ISO/IEC 25010:2023 standard address reliability as:

*Capability of a product to perform specified functions under specified conditions for a specified period of time without interruptions and failures.*

The definitions of reliability found in existing literature and the ISO/IEC 25010:2023 standard provide valuable insights into reliability in software systems. However, they overlook the inclusion of vendor reliability. Therefore, we find it advisable to incorporate vendor reliability into the definition, while also consolidating the strengths of the various definitions.

Based on the ISO/IEC 25010:2023 standard, all the definitions from the literature [9, 71, 73–75] and our interpretations, we suggest an overall definition of *reliability* to be:

*Reliability in a software ecosystem refers to the software's and vendor's capability to consistently perform specified functions accurately and without failure under stated conditions for a defined duration.*

Sub attributes:

| | |
|---|---|
| **Reliability** | Robustness |
| | Accuracy |
| | Availability (2x papers) |
| | Fault Tolerance |
| | Recoverability (2x papers) |
| | Maturity (2x papers) |
| | Reliability Compliance |

The definitions of reliability in existing literature and the ISO/IEC 25010:2023 standard offer valuable insights. However, we propose merging elements from these definitions and integrating the vendor's reliability into the definition. This refinement presents findings clearly, while also highlighting the importance of the vendor's reliability in meeting agreed-upon timelines. This inclusion is vital for the success of collaboration within the ecosystem and, consequently, the ecosystem itself.

NTNU

Norwegian University of
Science and Technology