2013 46th Hawaii International Conference on System Sciences

Improving decision support for software component selection through systematic cross-referencing and analysis of multiple decision criteria

Christoph Becker, Michael Kraxner, Markus Plangg, Andreas Rauber
Vienna University of Technology
{becker,kraxner,plangg,rauber}@ifs.tuwien.ac.at

Abstract

This article discusses opportunities for leveraging scale in cases of recurring scenarios of comparable decisions with multiple objectives in well-defined domains. Based on a software component ranking and selection method that uses utility analysis to separate objective information gathering and subjective assessment, we discuss challenges of decision making such as criterion complexity and evaluation effort.

We show that by systematically identifying criteria across cases, it becomes feasible to employ cross-referencing and quantitative assessment of decision criteria and criteria sets across scenarios and organizations to improve decision making efficiency and effectiveness. We present a method and tool that allows referencing decision criteria across cases and employs a set of impact factors for decision criteria and sets of criteria. We discuss the results of analyzing a series of real-world case studies in software component selection. We analyze the applications and implications of the method and its potential to improve decision making effectiveness and efficiency.

1. Introduction

The task of choosing a software component for a specific function in order to integrate it in a software system is a typical case of multi-criteria decision making that frequently occurs in Software Engineering.

Consider a decision maker with a set of components to fulfill a function in a software system, for example creating digital signatures on files. A number of decision factors will come into play such as functional suitability, security, performance efficiency, interoperability and costs. Some of these may pose conflicts: For example, increased security may come at the price of decreased performance efficiency or increased price. The decision maker has to follow a trustworthy and repeatable procedure to choose the component that best fulfills the objectives at hand.

The domain of component selection presents an interesting case of multiple criteria decision support systems (MCDSS) since it exhibits a number of peculiarities:

- A comparably large number of decisions of a very similar kind is made [1,4].
- The number of alternatives and decision criteria can be quite large. For example, [4] reports on a number of cases where between 30 and 50 decision criteria were used and hundreds of metrics were collected in each case.
- The decision criteria are rather well understood in terms of the facets and quality aspects that are evaluated. However, the individual assessment of each criterion's utility towards these aspects varies substantially among cases.

In these scenarios, the problem of eliciting, specifying, evaluating and weighing the criteria becomes challenging, and the complexity of making a choice is correspondingly high. Given the scale of the decision making problem, the primary goals for improving decision support are the decision makers' efficiency and effectiveness in reaching a choice on components.

In this article, we hence discuss opportunities for leveraging scale in MCDSS for component ranking and selection problems. We discuss the key questions that decision makers are facing and the challenges of decision making efficiency and effectiveness. Based on a systematic identification of criteria from real-world cases and the separation of objective evidence and subjective assessment through utility functions, we discuss a set of factors designed to assess the impact of criteria and sets of criteria across decision making scenarios. We show that by using standardized models to cross-reference and assess decision criteria and criteria sets across scenarios and organizations, a number of insights into decision making factors can be obtained which enable improvements to the decision support system.

This article is structured as follows. Section 2 discusses related work in the area of multiple criteria decision making and software component selection. Section 3 illustrates the particular challenges of the decision making scenario at hand and illustrates

1530-1605/12 $26.00 © 2012 IEEE
DOI 10.1109/HICSS.2013.263

IEEE computer society

opportunities for improvement. Section 4 outlines our method and step-wise analysis approach, while Section 5 applies the Goal-Question-Metric paradigm to define a set of impact factors for decision criteria and criteria sets. Section 6 discusses tool support and the application of our method to a number of real-world cases. Finally, Section 7 discusses implications and applications of the approach and outlines further steps.

2. MCDSS for component selection

Numerous approaches have been proposed for the general problem of software component evaluation and selection [1]. Most methods for component selection employ a variation of the standard five steps described in [14]:

1. Define criteria
2. Search for components
3. Shortlist candidates
4. Evaluate candidates
5. Analyze results and choose component

Frequently employed approaches for evaluating and selecting components include the usage of simple scoring and weighted sum approaches, the Analytic Hierarchy Process (AHP) [7], or iterative filtering. Others use methods based on utility analysis [6] to tackle the incommensurability of decision factors. In particular in cases of strict requirements on trustworthiness and reliable selection of components, evidence-based decisions using controlled testing are recommended [4]. A comprehensive overview of approaches is given in [1].

For the scenario of component selection, using goal-based requirements modeling and utility analysis is especially suitable for a number of reasons: The decision models strongly build on quality attributes that lend themselves to requirements engineering approaches; the anomaly of rank reversal [13] should be avoided; and the number of analytical steps that for example the application of the AHP requires is in many cases prohibitive [4].

Still, the problematic aspect of all approaches for component selection that can be considered trustworthy, i.e. evidence-based and formalized, is the high complexity and effort involved in creating suitable evidence. This begins with the unambiguous specification of criteria for quality attributes, which can be quite challenging [3], and extends to the evaluation of components, i.e. the process of assigning values to decision criteria.

Software quality models have provided a common language to high-level aspects of the selection problem. The ISO standard 25010 - 'Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models' [9] is based on the earlier ISO 9126 family. It defines a hierarchy of high-level quality attributes. SQUARE combines a revised quality model with evaluation procedures based on ISO 14598 [10]. ISO 25010 defines a product quality model that describes static properties of software and dynamic properties of the computer system, and a quality in use model that describes the outcome of interaction when a product is used in a particular context [9]. These models specify a number of well-defined characteristics and sub-characteristics. ISO 25020 defines requirements on the specification of software product quality criteria [2]. Earlier, Franch proposed a six-step method for defining a hierarchy of quality attributes for a specific domain in a top-down fashion [11].

Such standardized quality models provide unambiguous reference points by defining a top-down quality model for software systems. However, they are not decision models; they are independent domain models that describe and standardize quality attributes of the choices to be made. Moreover, the individual criteria and metrics that are used to evaluate the options according to these aspects are left to each decision maker. That means that for each quality attribute such as *functional correctness*, domain-specific decision criteria and a corresponding evaluation method still have to be defined.

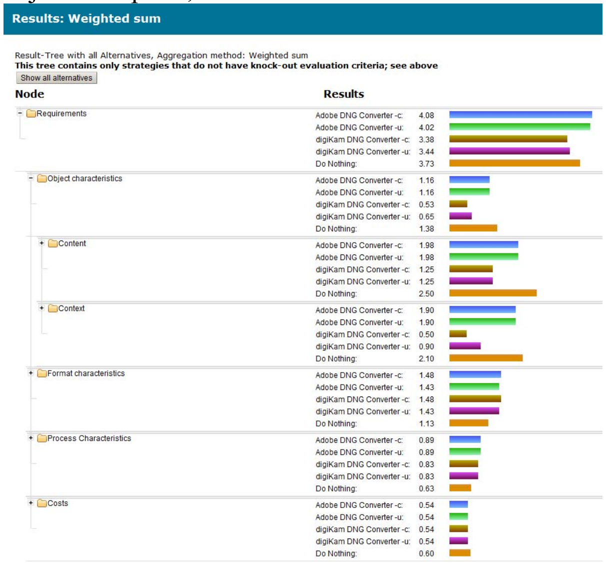
To separate objective evidence from the subjective assessment while addressing the incommensurability of decision criteria, utility functions can be used. In the framework discussed in this paper, a hierarchical model of objectives is built by the decision maker. The model uses utility functions for the leaf level of a tree and aggregation functions for overall scoring across the objective hierarchy.

Let s be a decision *scenario*, i.e. a description of key aspects of decision cases. Such a scenario may for instance be the concept of selecting a visualization component for business intelligence solutions; of selecting an encryption module for supply chain management systems; or of selecting file format conversion components in digital asset management systems.

Let *D* be a set of *decision cases* d_i, each case being a concrete instance of a scenario in which a decision maker specifies concrete decision criteria and evaluates actual components to choose the best fit.

Such a decision case will specify a number of *decision criteria* c_i, each with an objective aspect (such as the milliseconds required to perform a typical operation), a relative weighting against other criteria, and a utility function that calculates a normalized value between 0 and *max_util* for each objective measure. The extreme lower case of 0 denotes a reason for rejecting a candidate irrespective of its performance in other criteria, while *max_util* is the optimum score.

The objective aspect in turn can be linked to the abstract concept of a *criterion c* which represents the objective aspects, i.e. the metric used for evaluation.



Results: Weighted sum

Result-Tree with all Alternatives, Aggregation method: Weighted sum

This tree contains only strategies that do not have knock-out evaluation criteria; see above

Node	Results	
Requirements	Adobe DNG Converter -c:	4.08
	Adobe DNG Converter -u:	4.02
	digiKam DNG Converter -c:	3.38
	digiKam DNG Converter -u:	3.44
	Do Nothing:	3.73
Object characteristics	Adobe DNG Converter -c:	1.16
	Adobe DNG Converter -u:	1.16
	digiKam DNG Converter -c:	0.53
	digiKam DNG Converter -u:	0.65
	Do Nothing:	1.38
Content	Adobe DNG Converter -c:	1.98
	Adobe DNG Converter -u:	1.98
	digiKam DNG Converter -c:	1.25
	digiKam DNG Converter -u:	1.25
	Do Nothing:	2.50
Context	Adobe DNG Converter -c:	1.90
	Adobe DNG Converter -u:	1.90
	digiKam DNG Converter -c:	0.50
	digiKam DNG Converter -u:	0.90
	Do Nothing:	2.10
Format characteristics	Adobe DNG Converter -c:	1.48
	Adobe DNG Converter -u:	1.43
	digiKam DNG Converter -c:	1.48
	digiKam DNG Converter -u:	1.43
	Do Nothing:	1.13
Process Characteristics	Adobe DNG Converter -c:	0.89
	Adobe DNG Converter -u:	0.89
	digiKam DNG Converter -c:	0.83
	digiKam DNG Converter -u:	0.83
	Do Nothing:	0.63
Costs	Adobe DNG Converter -c:	0.54
	Adobe DNG Converter -u:	0.54
	digiKam DNG Converter -c:	0.54
	digiKam DNG Converter -u:	0.54
	Do Nothing:	0.60

Figure 1 Visualization of scores in one case

In each decision case, a set of *alternatives* a_i are evaluated against a set of decision criteria. For each alternative, an overall score can be computed by calculating the aggregated utility of all criteria. Figure 1 shows the high-level aggregation in our decision support tool for a real-world case in which 7 candidates were evaluated against 69 criteria in a 7-layer hierarchy. The overall score for an alternative a_i, $score(a_i,d_i)$, is the weighted sum of the criteria utilities aggregated across the hierarchy of objectives.

This approach combines objective evidence measured in specific scales, subjective assessment represented in explicitly defined scenario-specific utility functions, and relative weights across the goal hierarchy. As such, it is a flexible model, but it requires a profound understanding of the intricacies of decision making scenarios. Furthermore, a careful distinction between the key concepts of evidence, utility, and weighting is expected from the decision maker.

To support the decision maker in understanding uncertainty and analyze the effects of variation, common approaches to sensitivity analysis vary the weightings of attributes to determine the robustness of assigned weights [8]. However, solely assessing the robustness of weights focuses only on the choice and one aspect of the specification of criteria. As such, it fails to take into account a number of sources for errors:

1. It does not include the measurement uncertainty inherent in the process of gathering evidence for evaluation.
2. It does not address human error in decision making, in particular in the specification of criteria itself: Relevant factors might have been omitted or not specified correctly.
3. It does not address the dimension of utility, which might increase or decrease the impact of a measurement error, depending on the steepness of the utility function at the point of measurement.

3. Challenges and Opportunities

Decision makers in the described scenarios face a number of challenges.

- The quality and completeness of the specified decision criteria is often unclear.
- The effort required to specify criteria is high, since choosing decidable criteria for a desirable quality attribute is complex.
- The effort required to perform an evaluation for certain criteria may be very high. For example, evaluating performance efficiency or the functional correctness of software components requires controlled experimentation, and functional correctness in particular is sometimes very complex to evaluate. For larger numbers of criteria, the total effort required to evaluate options becomes substantial. For example, in the cases reported in [4], hundreds of evaluation steps were required to complete the decision and choose one component.

In order to better understand the scenario at hand, decision makers would require more specific insight and guidance from the decision support systems. This, however, requires us to answer a number of difficult questions:

1. What is the impact of a certain criterion on the decision? Would a change in its evaluation, i.e. in the objective evidence, change preference rankings on alternative solutions?

2. Considering a specific case: How critical was this criterion in other cases? Has it led to the exclusion of potential alternatives in similar cases?

3. What is the accumulated impact of a set of criteria on decisions in certain scenarios?

4. What is the minimum set of criteria that have to be considered in a given scenario? Can we remove criteria from our set to reduce evaluation effort without influencing the evaluation outcome? Are there any (sets of) decision criteria that are dominated?

It has to be noted that the goal of the decision maker in these cases is often not to discover the *perfect* solution, the optimal component, but instead to reach a sufficiently near-optimal decision in an effective and efficient manner. That means that the risk that there is a candidate outperforming the winning candidate will generally be considered much less harmful than the risk of the winning candidate performing suboptimal, i.e. failing on key criteria, and the cost of performing a full in-depth evaluation of alternatives that are unlikely to yield optimal results.

The cases we are referring to increasingly rely on standardized models for top-down classification of criteria, such as software quality models. These specify comprehensive taxonomies for recurring quality aspects of interest. Integrating such models opens up opportunities to leverage the scale of multiple related scenarios by linking data across scenarios through a standard quality model. This provides most value within a domain, but can also be useful across domains.

The next section will discuss a systematic 5-step approach to conduct a systematic assessment of decision criteria in order to enable improved tool support and increase the efficiency and effectiveness in the described scenarios.

4. Methodical approach

4.1 Overview

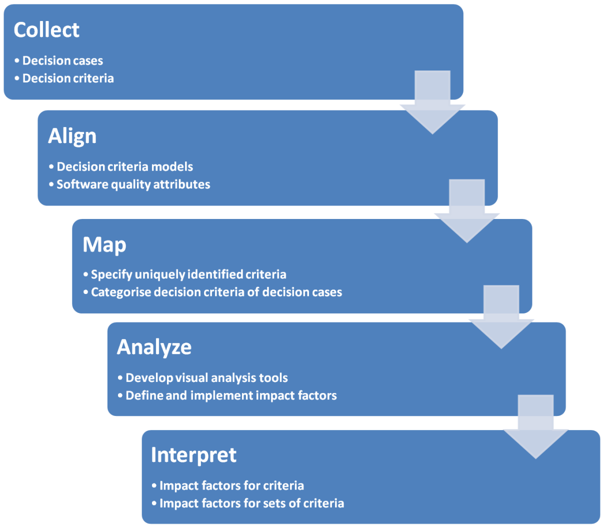

Collect
• Decision cases
• Decision criteria
Align
• Decision criteria models
• Software quality attributes
Map
• Specify uniquely identified criteria
• Categorise decision criteria of decision cases
Analyze
• Develop visual analysis tools
• Define and implement impact factors
Interpret
• Impact factors for criteria
• Impact factors for sets of criteria

Figure 2 Systematic analysis of decision criteria

Figure 2 describes the main steps of the methodical analysis approach presented in this article. Based on a collection of real-world case studies and the contained decision criteria, we aligned the criteria with standard quality and criteria models. This leads to a categorization and identification of sets of criteria. To allow quantitative analysis and assessment, we apply the Goal-Question-Metric (GQM) paradigm to develop a set of impact factors. These have to be applicable to criteria and sets of criteria. Section 5 will discuss this step in detail. Finally, we have developed a visual analysis tool to support decision makers.

4.2 Collect

The decision support framework we are building our work on is being actively used by a number of organizations. The starting point of the systematic classification step is provided by a set of 14 decision cases that evaluated a total of 51 components against a total of 631 decision criteria.

4.3 Align

In order to allow systematic identification and specification of criteria across decision cases, we have developed a framework for systematically identifying and cross-referencing quality attributes and decision criteria for arbitrary sets of decision scenarios. The framework is based on well-understood top-down taxonomies for classifying decision criteria, including ISO SQUARE. In each decision case, the metrics used for gathering objective evidence are mapped to uniquely identified criteria. Tool support enables decision makers in this mapping.

Decision criteria were thus mapped to standard models of software quality, business aspects such as costs, and domain-specific criteria definitions. This analysis phase corresponds to the process of structuring a reusable criteria catalogue as described in [3], where quality models and actual decision criteria are aligned and potential specification conflicts have to be reconciled. This led to a criteria catalogue with currently about 400 uniquely identified criteria. These are represented in an OWL[1] ontology and published conforming to Linked Data principles[2], so that decision support tools can easily reference them.

4.4 Map

Based on the reconciled and fully specified criteria catalogue, decision criteria of all decision cases were mapped to standardized criteria specifications. This involved in some cases minor, behavior-preserving refinements of the specification of decision criteria. In the 14 decision cases, 92% of the criteria were mapped to standard criteria in the criteria catalogue: 579

[1] http://www.w3.org/TR/owl-ref/

[2] http://www.w3.org/DesignIssues/LinkedData.html

decision criteria are referencing 368 unique criteria. The remaining decision criteria were considered too specific to merit inclusion in the catalogue. This corresponds to entries in a "non-reusable criteria catalogue" as described in [3].

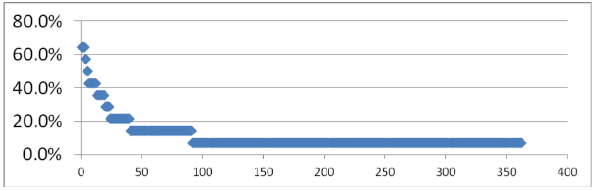


Figure 3 Distribution of criteria mappings

Figure 3 shows the frequency distribution of criteria across the set of 14 decision cases. As can be expected, the frequency distribution roughly corresponds to a power law distribution. It shows a long tail of rarely used criteria and a corresponding number of very frequently used criteria. 24 criteria are used in more than 25% of the decision cases, with only three criteria being used in two thirds of all decisions, while 270 criteria were referenced only once in this set of cases.

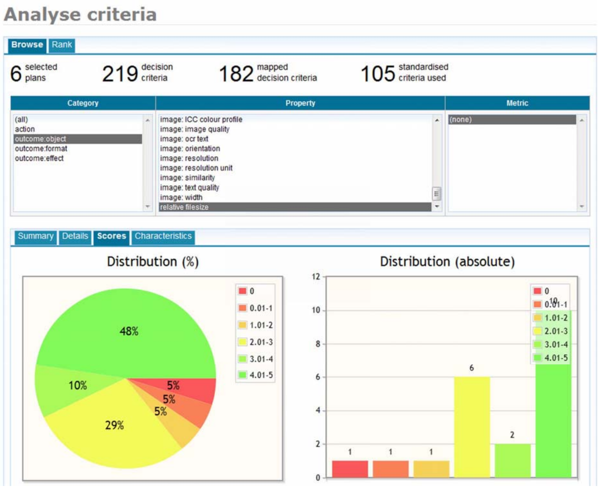
4.5 Analyze

The development phase comprises a set of visual tools and a set of quantitative indicators for criteria.

The web-based decision support tool which is the background of this work is maintained as a software service and currently has around 800 registered users. A small number of these have conducted real-world business decisions using the framework and tool. The entire software suite, including the analytical module discussed here, is freely available and published under an open license on github.[3]

The analysis module is an extension of this system. It allows domain experts to browse criteria according to categories and analyze the effects of criteria across different cases in real-time to provide direct insight to subject matter experts and support flexible analysis of the criteria catalogue. The tool shows all cases in anonymized form to the decision maker.

Figure 4 shows the interface to browse criteria. We see that the analysis set focuses on 6 closely related decision cases, in which 219 criteria are used. Of these, a total 182 decision criteria are referencing a set of 105 standardized criteria (out of the several hundred in the criteria catalogue). The bottom part of Figure 4 visualizes the distribution of scores in the selected scenarios for one criterion. In 5% of the cases, a worst-case score of 0 was provided, which constitutes an unacceptable performance.



Analyse criteria

Browse | Rank

6 selected plans
219 decision criteria
182 mapped decision criteria
105 standardised criteria used

Category	Property	Metric
(all) action outcome:object outcome:format outcome:effect	image: ICC colour profile image: image quality image: ocr text image: orientation image: resolution image: resolution unit image: similarity image: text quality image: width relative filesize	(none)

Summary | Details | Scores | Characteristics

Figure 4 Criteria used in multiple scenarios

A closer look into the variance across decision cases is provided in Figure 5, which shows the visualization of different stakeholders' preferences towards a numeric metric. The curve on the bottom depicts three decision maker's utility functions, while the top distribution shows the different scores achieved by candidate software components. More than half of the components gained top scores, while none was graded with the worst-case score of 0. The utility curves also visually illustrate the importance of accounting for individual preferences in the robustness analysis: The same objective change can lead to drastically different fluctuations in scores across decision cases.

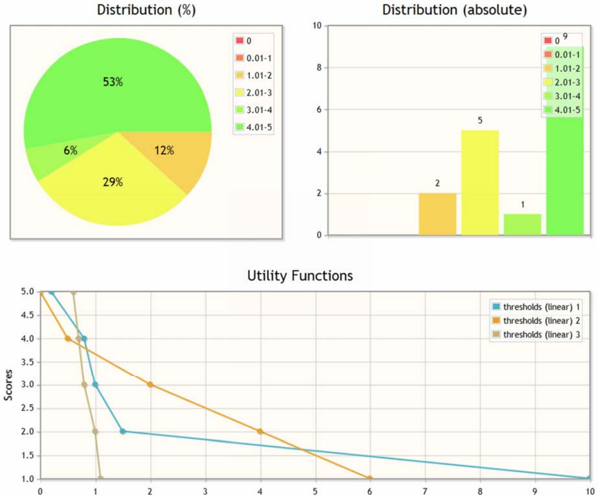


Figure 5 Different utilities for one criterion

4.6 Interpret

This visual analysis allows for interesting insight into the variance across scenarios. However, to answer the questions raised above, we need more specific

[3] http://github.com/openplanets/plato

indicators per criterion. The next section will define such indicators. In Section 6 we will discuss their application to a set of real-world case studies and illustrate their integration in the decision support system.

5. Impact factors for decision criteria

In search for realistic, relevant and representative indicators, this section applies the Goal-Question-Metric approach (GQM) [12] and defines a number of impact factors for single criteria and groups of criteria.

The questions raised above require an accumulated assessment of the impact of arbitrary sets of criteria over sets of cases, where each criterion may appear in a number of cases. To achieve this, we will define impact factors for single criteria and sets of criteria. These factors need to reflect

- the usage frequency of a criterion in comparable scenarios,
- the average weight of the criterion in scenarios where it appears, and
- a criterion's sensitivity, i.e. the extent to which the utility scores of decisions including the criterion change when the evaluation facts change. For this, we need to consider the objective evidence collected and the utility functions defined in each decision case.

Let $C = \{c_1, c_2, ...c_n\}$ be a non-empty set of criteria and $D = \{d_1, d_2, ...d_m\}$ the set of decision cases considered – for example, all decisions selecting file format conversion components in digital asset management systems. Then for a criterion $c \in C$, D_c is the set of decisions using c. A key aspect to consider is the *potential output range por(c,d)* of a criterion c in a decision case d, i.e. the maximum change it causes on the overall score of alternatives in a decision scenario. This change results from its utility function $u_{c,d}$ and weighting $w_{c,d}$ and the potential range of input values allowed. We are furthermore interested in its *actual output range aor(c,d)* resulting from the application of values obtained as objective evidence. The latter is given by the weighted difference between the lowest and highest result of the utility function applied to the actual evaluation values $v_c \in d$: $aor(c, p) = w_{c,d} \times (\max(u_{c,d}(v_{c,d})) - \min(u_c,p(v_{c,d})))$, with $aor(c, d) \leq por(c, d) \forall c \in C, d \in D$. Finally, the *relative output range* $ror(c, p) = aor(c,p) / por(c,p)$ measures the variation of scores within allowed boundaries.

Finally, a discrete effect is presented by the usage of 0 as a mechanism for filtering unacceptable aspects: For each criterion, we are interested in how many alternatives are rejected for unacceptable performance.

A number of combinations of these aspects are possible. However, only a select few of these will be meaningful to answer concrete questions. We will hence start from the decision making goals and discuss specific indicators and metrics as they can be derived from these goals.

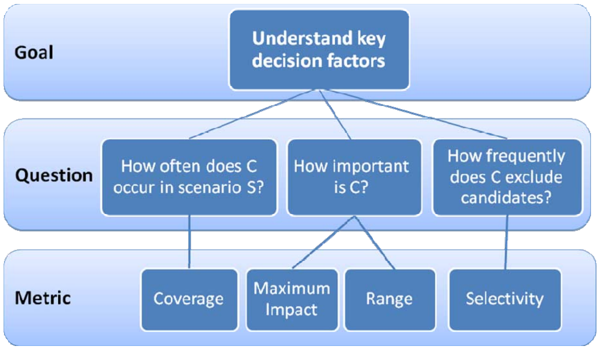

Goal
Understand key decision factors
Question
How often does C occur in scenario S?
How important is C?
How frequently does C exclude candidates?
Metric
Coverage
Maximum Impact
Range
Selectivity

Figure 6 Metrics for key decision factors

Figure 6 starts with the key goal of understanding key factors in a well-defined decision making scenario. The key questions relate to the frequency of occurrence of criteria, their output range, and the question of selectivity: In how many cases are components excluded because they fail a criterion?

We define **Coverage** as the percentage of decision cases using at least one of the criteria in C, i.e. $|D_c|/|D|$.

Maximum Impact is defined as the maximum sum of *actual output ranges* encountered across all decision cases in D_c, while **Range** denotes the average impact, i.e. the sum of *actual output ranges* in D_c divided by $|D_c|$.

Selectivity is the percentage of alternatives excluded, i.e. the percentage of pairs *(alternative, d_i), with $d_i \in D_c$*, for which at least one of the criteria in C resulted in a utility score of 0.

Considering Figure 7, the quest for increasing efficiency by finding minimum representative criteria sets requires the introduction of **Significance**. This addresses the question whether a set of criteria serves as a differentiating factor at all or whether it is dominated. We can calculate this in a straightforward way by iterating through all criteria for each decision case in the following manner:

1. If a criterion c_i has *selectivity* $(c_i) > 0$, it is significant in d_i, and we continue with the next criterion.
2. If *selectivity* $(c_i) = 0$ in the considered case d_i, we set the winning candidate's utility score to the minimum achievable utility and all other candidate's score to the maximum. If this causes a difference in the final ranking of candidates, the criterion is considered significant.

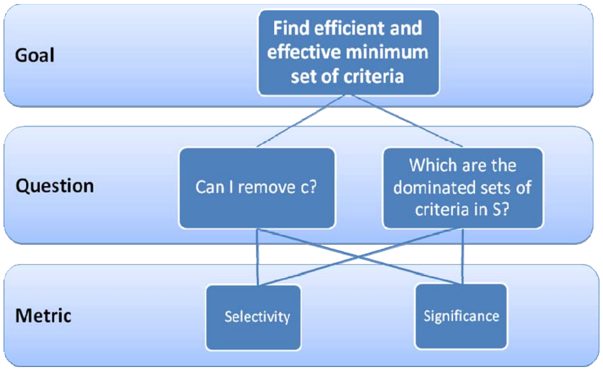

Goal
Find efficient and effective minimum set of criteria
Question
Can I remove c?
Which are the dominated sets of criteria in S?
Metric
Selectivity
Significance

Figure 7 Metrics for efficient criteria sets

In other words, a criterion is considered significant if and only if its selectivity is positive or if setting the criterion's score to the lowest extreme for the winning candidate and to the highest extreme for the other candidates causes the overall preference ranking of candidates in the decision case to change. This definition is intentionally defensive to prevent exclusion of relevant criteria.

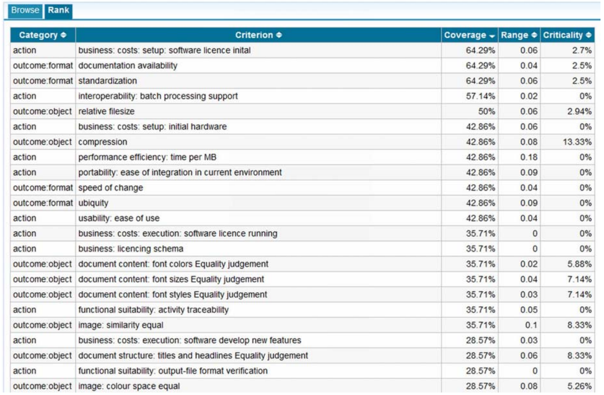
By exclusion, we thus obtain an initial set of criteria that are considered *insignificant* on their own. Based on this, we iterate and build larger sets of criteria which, when combined, are still considered insignificant. The intersection of these sets across a set of decision cases is the set of dominated sets of criteria.

Finally, we introduce **measurement robustness**, which indicates the degree to which measurement error causes a change in scores. This is calculated in two distinct ways per criterion:

- For numeric values, robustness is the percentage that we need to change the measurement result on the overall winning candidate to make the winning candidate lose its winning rank.
- For ordinal values, robustness is the percentage of possible measurement results for the overall winning candidate that would *not* cause the winning candidate to lose its winning rank. For example, in the case of possible ordinal values *good*, *bad*, or *ugly* with the winning candidate scoring *good,* we might find that the winner being evaluated as *bad* does not make it lose its overall best score in d_i, but rating it as *ugly* changes the ranking in d_i,. The corresponding robustness of this criterion for the decision case d_i is *0,66*.

6. Analysis of case studies

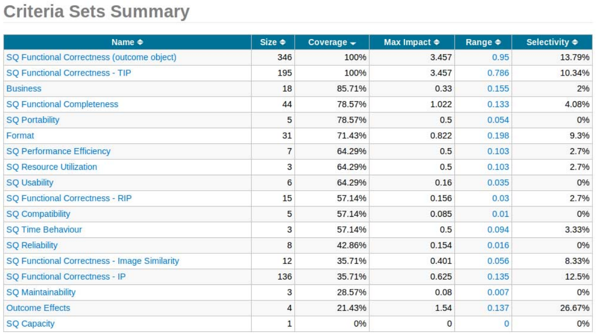
To discuss the applicability, relevance and usefulness of the proposed analysis and the indicators derived from the analysis goals, we discuss the application of the proposed impact factors to a set of 14 decision cases in component selection. These cases were selected for their complete and high-quality documentation of the decision making process and the full specification of decision criteria.



Category ◆	Criterion ◆	Coverage ▾	Range ◆	Criticality ◆
action	business: costs: setup: software licence inital	64.29%	0.06	2.7%
outcome:format	documentation availability	64.29%	0.04	2.5%
outcome:format	standardization	64.29%	0.06	2.5%
action	interoperability: batch processing support	57.14%	0.02	0%
outcome:object	relative filesize	50%	0.06	2.94%
action	business: costs: setup: initial hardware	42.86%	0.06	0%
outcome:object	compression	42.86%	0.08	13.33%
action	performance efficiency: time per MB	42.86%	0.18	0%
action	portability: ease of integration in current environment	42.86%	0.09	0%
outcome:format	speed of change	42.86%	0.04	0%
outcome:format	ubiquity	42.86%	0.09	0%
action	usability: ease of use	42.86%	0.04	0%
action	business: costs: execution: software licence running	35.71%	0	0%
action	business: licencing schema	35.71%	0	0%
outcome:object	document content: font colors Equality judgement	35.71%	0.02	5.88%
outcome:object	document content: font sizes Equality judgement	35.71%	0.04	7.14%
outcome:object	document content: font styles Equality judgement	35.71%	0.03	7.14%
action	functional suitability: activity traceability	35.71%	0.05	0%
outcome:object	image: similarity equal	35.71%	0.1	8.33%
action	business: costs: execution: software develop new features	28.57%	0.03	0%
outcome:object	document structure: titles and headlines Equality judgement	28.57%	0.06	8.33%
action	functional suitability: output-file format verification	28.57%	0	0%
outcome:object	image: colour space equal	28.57%	0.08	5.26%

Figure 8 Most frequently used criteria

Figure 8 shows the most frequently used criteria in the case study set, ordered by descending coverage. It shows that 19 criteria are used in at least a third of the cases. The values in the column *Range* show that many of these criteria have minimal influence on the final ranking of candidates, which points to potential for optimizing decision processes. However, a number of criteria that have minimal output range are clearly significant, as shown by their selectivity.



Criteria Sets Summary

Name ◆	Size ◆	Coverage ▾	Max Impact ◆	Range ◆	Selectivity ◆
SQ Functional Correctness (outcome object)	346	100%	3.457	0.95	13.79%
SQ Functional Correctness - TIP	195	100%	3.457	0.786	10.34%
Business	18	85.71%	0.33	0.155	2%
SQ Functional Completeness	44	78.57%	1.022	0.133	4.08%
SQ Portability	5	78.57%	0.5	0.054	0%
Format	31	71.43%	0.822	0.198	9.3%
SQ Performance Efficiency	7	64.29%	0.5	0.103	2.7%
SQ Resource Utilization	3	64.29%	0.5	0.103	2.7%
SQ Usability	6	64.29%	0.16	0.035	0%
SQ Functional Correctness - RIP	15	57.14%	0.156	0.03	2.7%
SQ Compatibility	5	57.14%	0.085	0.01	0%
SQ Time Behaviour	3	57.14%	0.5	0.094	3.33%
SQ Reliability	8	42.86%	0.154	0.016	0%
SQ Functional Correctness - Image Similarity	12	35.71%	0.401	0.056	8.33%
SQ Functional Correctness - IP	136	35.71%	0.625	0.135	12.5%
SQ Maintainability	3	28.57%	0.08	0.007	0%
Outcome Effects	4	21.43%	1.54	0.137	26.67%
SQ Capacity	1	0%	0	0	0%

Figure 9 Criteria sets and impact factors

Taking a more high-level look at quality aspects, the analysis tool allows the specification of named sets of criteria. Figure 9 shows criteria sets and their impact factors. The sets defined here stem from a combination of standard SQUARE quality attributes such as functional completeness with domain-specific requirements. For example, the set *format* contains indicators about output formats produced by file format conversion tools.

It can be seen that a number of quality aspects such as portability, usability and maintainability of software components are considered in many cases, but have negligible impact. Business factors (which most importantly include costing and licensing) have no selectivity, but provide for considerable variation in scores.

For some sets, there is a considerable difference between the maximum impact caused in one particular decision case and the range, i.e. the average impact across all decisions covered, as visualized in Figure 10. This stems from the variation in detailed aspects of relevance in each scenario and needs to be considered carefully, since using average impact factors will be overly simplistic.

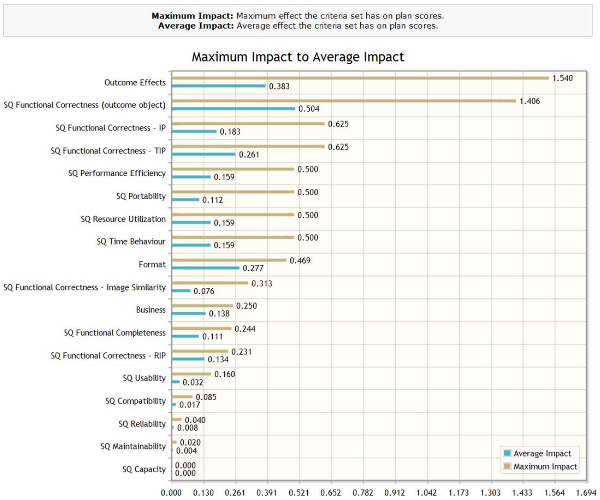


Figure 10 Average and maximum impact for criteria sets diverge substantially

Considering observations such as the negligible impact of certain criteria as evidenced in Figure 9 and the considerable number of criteria involved, two natural options arise to improve decision making efficiency: We can reduce the set of criteria evaluated or improve automation support in evaluating criteria in automated ways.

We observed that a number of frequently considered quality aspects had negligible impact and zero selectivity. These were in particular portability, capacity, reliability, usability, maintainability and compatibility. The question arises whether these aspects had in fact any significance in the scenario. Thus searching for dominated sets, we focus on a selected homogeneous subset of 6 decision cases. Applying the heuristic to calculate significance that was described in Section 5 provides interesting results:

- Of the abovementioned criteria sets with the lowest impact, none is fully dominated, although single criteria of these sets are frequently dominated.
- Of the 105 criteria considered in the six cases, 31 are in fact dominated across the set.

To evaluate the value of increasing automation, the cost-benefit relation has to be considered, since automated measures are sometimes expensive to provide.

As illustrated in Figure 11, we thus have to combine an analysis of coverage and significance of criteria with the costs of measuring, the possibility to verify that automated measures are correct, and the actual variation occurring in the real world as an indicator of how likely changes are to occur. Finally, average measurement robustness provides a valuable input to specify the required precision of measures: If a measurement error of 10% has no implications on decisions, investment into measurement devices with <1% precision will not be necessary.

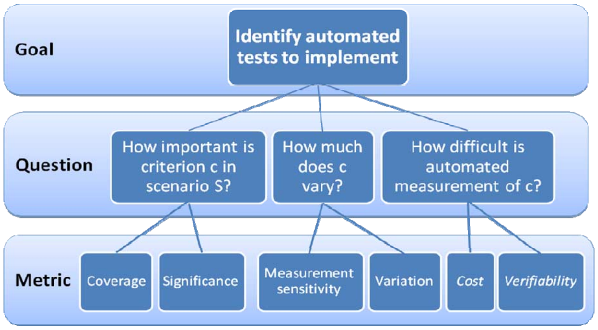

Goal
Identify automated tests to implement
Question
How important is criterion c in scenario S?
How much does c vary?
How difficult is automated measurement of c?
Metric
Coverage
Significance
Measurement sensitivity
Variation
Cost
Verifiability

Figure 11 Metrics for automation efforts

Analyzing the variation and measurement sensitivity of the set of cases shows interesting insights. Figure 12 shows the variation of the 93 criteria that are used more than once. It can be seen that there is a broad distribution across criteria, and a substantial fraction of the criteria cause the full range of utility scores they can produce.

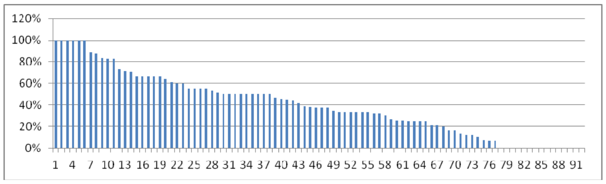


Figure 12 Utility variation of frequent criteria

However, very few single criteria have decisive sensitivity to measurement error in more than one decision case: Significance varies substantially between cases, so that conclusions across the board cannot be drawn. To provide a more specific detailed example, consider *relative filesize*, a criterion of general relevance in content management systems where conversion to smaller file sizes can save storage space and costs. The criterion is measured as a real number denoting a factor: *1.0* thus means that a file resulting from a conversion has the same size as the input file. The criterion is used in 7 cases with a variation of 44% of the potential utility range. Robustness varies from 57% to 400%, with one very robust case: In one decision case, even very large file sizes would not change the preference for the chosen component due to clear advantages in other aspects and the low range of the criterion.

7. Discussion and Outlook

In this paper we discussed challenges and opportunities to use particular characteristics of scale in decision making scenarios for component selection purposes. We outlined the characteristics of the problem space and discussed key questions arising from the perspective of decision makers; domain experts; and decision support systems. Our contribution builds on an existing decision support framework. We formalized quality criteria so that they can be cross-referenced and analyzed across scenarios. We specified a number of metrics for the quantitative evaluation of decision criteria and sets of criteria, and illustrated their application to a set of real-world decision cases.

Returning to the questions posed in Section 3, we can conclude that the impact factors can provide meaningful answers to decision makers:

1. Calculating a criterion's selectivity, significance and measurement robustness in a decision case enables us to understand its impact on a specific decision.
2. Assessing a criterion's selectivity and significance across cases allows us to understand its criticality across comparable decision scenarios.
3. Calculating accumulated coverage, range and selectivity of a set of criteria in a scenario enables us to arrive at a better understanding of the critical success factors of candidate choices.
4. Finally, assessing selectivity and significance supports us in reducing criteria sets to their key elements.

The question arises in how far these metrics can be in fact useful to improve decision making efficiency and effectiveness. Based on our analysis, we can draw the following observations:

- Through increased formalization and unambiguous specification, the semantics of criteria become clear and are documented more transparently. Correspondingly, the danger of incomplete and ambivalent criteria specification drops. This is supported by allowing decision makers to analyze the anonymized preference structures of others and gain a better understanding into the decision space at hand, as shown in Figures 4 and 5.
- The combination of coverage, range and selectivity (illustrated in Figure 6) can inform which aspects are considered significant by decision makers who are domain experts. By deepening the understanding of key decision factors of particular domains, tool support and quality checks can be tailored to specific scenarios. The quantitative assessment of specific aspects of standardized quality models can further improve communication between solution providers and procurement, since clear performance statistics for recognized and well-understood priorities can be communicated.
- The combination of selectivity and significance (illustrated in Figure 7) can be used to discover sets of criteria that are dominated within the context of a scenario. By understanding the difference between selective, significant and insignificant criteria, we can further optimize decision making efficiency in a number of ways. For example, ordering the criteria for the evaluation procedure by decreasing significance and impact can allow the decision support system to cut off evaluation for candidates that are outperformed by other options. Furthermore, this provides a mechanism to prioritize automated measures to be developed.
- Finally, the combination of significance, variation and robustness, and costs of automating decision criteria measurement provides valuable input for analyzing the cost-benefit relation of automation. Assessing the costs of providing automated measures will still be challenging. However, the combined application of these metrics should provide valuable guidance for focused improvement efforts.

The analysis in this paper is not concerned with the different hierarchies that decision makers might use to structure the standard quality attributes according to their understanding. The impact factors of criteria are focused on the unambiguously defined criteria themselves. However, the specification of arbitrary sets of criteria allows us to group relevant sets such as software quality characteristics together, independently of their grouping in each decision case, and assess their cumulative impact across cases.

The impact factors can support a reduction of effort and complexity in component selection activities in a number of ways. For example, they can be used to reduce the number of alternatives that are evaluated in depth, remove dominated sets, or filter candidate components based on correlating goals and constraints with documented knowledge as well as experience shared by other decision makers.

The assessment of decision factors demonstrated here can also be beneficially combined with existing complementary approaches to increase efficiency in component selection. For example, the COTS Acquisition Process (CAP) presented in [15] is based on the premise that measurement of all applicable criteria is too difficult and expensive and aims at increasing efficiency in the selection process by minimizing the actual measurements taken. Knowing which criteria are likely to have the largest impact and which may be dominated by other sets can greatly increase the effectiveness of such approaches.

Finally, it is sometimes difficult to recognize hidden criteria that may be relevant to a scenario. Integrating aspects such as co-occurrence and correlation should support us in answering questions such as "Has the decision maker covered all relevant aspects? Are there any overlooked criteria that are related to included aspects, but have been missed? Can these be critical?"

By systematically analyzing co-occurrence, correlation and impact of decision criteria across cases, it should be possible to integrate recommender systems into the decision making workflow that can provide increased guidance and warn decision makers of potential risks and opportunities based on others' experiences.

Current work is geared towards quantitative baseline assessment of decision making effort and the introduction of targeted improvements in decision support that can be evaluated objectively for efficiency and effectiveness. This will include the integration of proactive recommendations and warning heuristics for relevant hidden criteria based on co-occurrence and correlation.

References

[1] Anil S. Jadhav and Rajendra M. Sonar. Evaluating and selecting software packages: A review. Information and Software Technology, 51(3):555–563, 2009.

[2] ISO. Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Measurement reference model and guide (ISO/IEC 25020:2007). International Standards Organisation, 2007.

[3] Juan Pablo Carvallo, Xavier Franch, and Carme Quer. Determining criteria for selecting software components: Lessons learned. IEEE Software, 24(3):84–94, May-June 2007.

[4] Christoph Becker and Andreas Rauber. Improving component selection and monitoring with controlled experimentation and automated measurements. Information and Software Technology, Volume 52, Issue 6, June 2010, Pages 641-655

[5] J. Figueira, S. Greco, and M. Ehrgott. Multiple criteria decision analysis: state of the art surveys. Springer Verlag, 2005

[6] R. L. Keeney and H. Raiffa. Decisions with multiple objectives: preferences and value tradeoffs. Cambridge University Press, 1993.

[7] Thomas L. Saaty, How to make a decision: the analytic hierarchy process, European Journal of Operational Research 48 (1) (1990) 9–26.

[8] J. Butler, J. Jia, and J. Dyer. Simulation techniques for the sensitivity analysis of multi-criteria decision models. European Journal of Operational Research, 103(3):531 – 546, 1997.

[9] ISO/IEC. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (ISO/IEC 25010). International Standards Organisation, 2011

[10] ISO. Information technology – Software product evaluation – Part 1: General overview (ISO/IEC 14598-1:1999). International Standards Organization, 1999.

[11] X. Franch and J. Carvallo. Using quality models in software package selection. IEEE Software, 20(1):34–41, Jan/Feb 2003

[12] Basili, V.R.; Caldiera, G.; Rombach, H.D. The Goal Question Metric Approach, In: Encyclopedia of Software Engineering, Vol. II, September, pp. 528-532, 1994.

[13] V. Belton and T. Stewart, Multiple Criteria Decision Analysis: An Integrated Approach. Kluwer, Boston, MA, 2002.

[14] Abdallah Mohamed, Guenther Ruhe, Armin Eberlein, COTS selection: past, present, and future, in: Proceedings of the ECBS '07, 2007, pp. 103–114.

[15] M. Ochs, D. Pfahl, G. Chrobok-Diening, B. Nothhelfer-Kolb, A method for efficient measurement-based COTS assessment and selection method description and evaluation results, in: Proceedings of the 7th International Symposium on Software Metrics 2001, METRICS 2001, 2001, pp. 285–296.