Computer Standards & Interfaces 97 (2026) 104122

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A multi-criteria process for IT project success evaluation–Addressing a critical gap in standard practices

João Carlos Lourenço [a], João Varajão [b,*]

[a] CEGIST, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
[b] Centro ALGORITMI, Universidade do Minho, Campus de Azurém, 4804-533 Guimarães, Portugal

ARTICLE INFO

Keywords:
Project success
Project evaluation
Multi-criteria evaluation
MACBETH
Process
Methodology

ABSTRACT

The evaluation of project success is widely recognised as valuable for improving IT (Information Technology) project performance and impact. However, many processes fail to adequately address the requirements for a sound evaluation due to their inherent complexity or by not complying with fundamental practical and theoretical concepts. This paper presents a process that combines a problem structuring method with a multi-criteria decision analysis approach to evaluate the success of IT projects. Put into practice in the context of a software development project developed for a leading global supplier of technology and services, it offers a new way of creating a model for evaluating project success and tackling uncertainty, bringing clarity and consistency to the overall assessment process. A strong advantage of this process is that it is theoretically sound and can be easily applied to other evaluation problems involving other criteria. It also serves as a call to action for the development of formal standards in evaluation processes. Practical pathways to achieve such standardization include collaboration through industry consortia, development and adoption of ISO frameworks, and embedding evaluation processes within established maturity models. These pathways can foster consistency, comparability, and continuous improvement across organizations, paving the way for more robust and transparent evaluation practices.

1. Introduction

The sustainable success of virtually any organisation is strongly associated with the success of its projects [1]. A key factor for project success is that project managers clearly understand what success means [2], which is usually not the case [3]. Despite different notions about what constitutes "project success" and the many criteria that can be used for evaluation (e.g., cost, time, and performance, among others) [4], a project must satisfy its clients to be considered successful [5–8].

Given the importance and complexity of the evaluation of projects, companies should define and implement systematic processes for evaluating success to improve project management performance and the impact of deliverables [9]. However, despite the models and techniques that are currently available for assessing project success, they are typically challenging to implement for a variety of reasons, notably the complexity caused by using multiple and often conflicting objectives (e.g., minimise cost and maximise quality), the scarcity of empirical studies reporting their genuine use in projects [10], and the fact that practices employed in companies are generally informal and simplistic [11].

Additionally, several errors identified by decision analysis literature [12,13] are often made, generating meaningless project success evaluations [14]. Some common mistakes involve not including relevant criteria in the evaluation model, not distinguishing the performance of a project from its value, assigning weights to evaluation criteria without considering the ranges of variation of their performance scales, and making calculations that violate measurement scales' properties. In other words, such evaluations are inconsistent with multi-attribute value theory (MAVT) and value measurement foundations.

Considering these limitations, this research proposes a process that combines a problem structuring method with a multi-criteria approach for evaluating the success of information technology (IT) projects supported by a real-world case. This process was developed and applied in the context of a project of GlobalSysMakers (for confidentiality reasons, the name of the company herein is anonymized), a leading global supplier of technology and services.

In the GlobalSysMakers project, the need for a new process arose because the project management team felt that the scoring model initially defined for success assessment, while helpful, lacked accuracy.

* Corresponding author.
E-mail address: varajao@dsi.uminho.pt (J. Varajão).

https://doi.org/10.1016/j.csi.2025.104122
Received 12 August 2025; Received in revised form 7 November 2025; Accepted 23 December 2025
Available online 24 December 2025

Following an appraisal of several methodological alternatives, a new multi-criteria approach combined with a problem structuring method was shown to be the best solution, providing the required precision and transparency to the process, along with a better understanding of the real meaning of the relative importance of each evaluation criterion. This paper describes the process developed in detail so that it can be replicated in other projects. Also, the results are presented and discussed, including contributions to theory and practice.

The proposed process, which combines a problem structuring method with a multi-criteria approach for evaluating IT project success, offers several theoretical implications. First, it advances the conceptualization of project success by integrating both subjective stakeholder perspectives and objective performance criteria, addressing the multidimensional and context-dependent nature of success in IT projects. Second, it contributes to decision theory and project management literature by demonstrating how problem structuring methods—typically underutilized in IT evaluation—can enhance the clarity and relevance of criteria selection and prioritization. Third, the integration of these methodologies provides a foundation for developing more robust, transparent, and adaptable evaluation frameworks, which can inform future theoretical models and empirical studies. Ultimately, this research supports the movement toward standardization by offering a replicable and theoretically grounded process that can be refined and generalized across different organizational and project contexts.

The remainder of this paper is organised as follows. Section 2 briefly reviews previous related work on project evaluation methods, cases, and multi-criteria evaluation methods. Section 3 describes the case context and the development of the success evaluation model using a process that combines a problem structuring model with a multi-criteria decision analysis approach. Section 4 discusses the results obtained. Finally, Section 5 presents the conclusions and avenues for further work.

2. Previous related work

2.1. Success of projects

Evaluation can be defined as the assessment and analysis of the efficiency and effectiveness of the project's activities and results. The evaluation looks at what is planned to do, what has been achieved, and how it has been achieved [15]. Kahan and Goodstadt [16] conceive evaluation as a set of questions and methods properly articulated to review processes, activities, and strategies to achieve better results. Therefore, the purpose of an evaluation is not just to find out what happened but to use that information to make the project better [17,18].

There are several evaluation approaches in the literature, some considerably complex regarding their practical operationalisation and use. Varajão et al. [10] present a comprehensive review of models and methods for evaluating information systems project success. Some examples are described and analysed next.

Bannerman and Thorogood [19] propose a framework for defining IT project success that provides a common language for communication and compares what stakeholders perceive as important. The authors list the criteria that should be used to assess the success of a project within five domains (process, project management, product, business, and strategy). However, they do not explain how to consider these domains and criteria together.

Barclay and Osei-Bryson [20] describe a structured framework named Project Objectives Measurement Model (POMM) to identify the criteria for evaluating an information system (IS) project and assigning a performance measure to each criterion. POMM applies *value-focused thinking* principles [21] and *goal question metric* methods [22]. An illustrative case is presented in which the importance of each criterion is directly assessed using an average of the stakeholders' answers based on a 5-point Likert scale. However, despite its virtues, this operation is neither quantitatively nor substantively meaningful [23], respectively, because a Likert scale is an ordinal scale [24,25] and averaging the weights of several stakeholders without a discussion obliterates their individual differences [26]. Additionally, the "importance of the criteria" should consider their respective performance ranges; otherwise, the resulting weights would be arbitrary [27].

Basar [28] proposes a methodology to evaluate the performance of IT projects in a fuzzy environment. She first identifies the evaluation criteria using the balanced scorecard method. Second, she determines the criteria weights with expert judgments and hesitant fuzzy weights. Then, the weights are used to evaluate the performance of IT projects in a Turkish company. The weighting process described in this paper is difficult for a non-expert evaluator to understand. Additionally, the quantitative performances of projects on the criteria are systematically normalised to scores between 0 and 1 with a linear transformation that may not correspond to the preferences of evaluators (which may be non-linear). The paper does not explain how to address the evaluation of the qualitative criteria.

Ismail [29] applies the Delphi method and conducts a seminar with experts to identify a construction project's potential evaluation criteria and group them into clusters. A relative importance index is calculated for each criterion with a weighted average of the responses to a survey expressed on a Likert scale. In a subsequent step, the experts 1) reduced the number of clusters and criteria and 2) assigned the same weight to the latter. Then, a priority index was calculated for each criterion with the Priority Evaluation Model (PEM) [30], which combines the "satisfaction" rate (assigned by the experts) and the "importance" of the criterion. The overall project success is obtained with a weighted sum of the averages of the priority indexes obtained on each cluster and the clusters' weights. However, the paper does not explain how these weights were assessed. Additionally, the Likert scale classifications cannot be used for calculating averages or other arithmetic calculations.

Nguvulu et al. [31] use a *Deep Belief Network* (DBN) to evaluate eight IT projects' performances after training the DBN with five projects of 12 months duration. The DPN automatically assigned weights and scores to the criteria, considering possible interactions between them. The authors stress the advantage of this approach by not considering human subjectivity. However, from our point of view, this is a weakness because the subjective preferences of project managers, clients, and other stakeholders should be considered in an evaluation process to avoid arbitrary results generated by inadequate analytical approaches.

Wohlin and Andrews [32] apply principal component analysis and subjective evaluation factors to estimate which projects are successful or unsuccessful out of a set of projects. This statistical approach may be used to identify key project characteristics, but it does not allow for evaluating the project's success according to stakeholders' preferences.

Yan [33] suggests the combined use of the *balanced scorecard* (BSC) [34], the *Analytic Hierarchy Process* (AHP), and the *Fuzzy Comprehensive Analysis method* (FCA), respectively, to construct a performance criteria system, assess the criteria weights, and obtain an overall evaluation score. The author explains how to obtain the performance criteria system, but does not explain the weighting and scoring components.

Yang et al. [35] apply a multi-criteria model for evaluating a software development project's success using the *Analytical Network Process* (ANP) [36] to assess the criteria weights at several hierarchical levels. The scores of a project on a given criterion were obtained by calculating the average of the scores assigned by five experts using a 5-point Likert scale. Note that, as mentioned above, averages should not be calculated with ordinal scales. In addition, ANP is based on AHP, a method with known issues that affect the validity of the criteria weights (see, e.g., [37–39]).

Section 2.2 reviews important concepts and methods related to multi-criteria evaluation that are needed to create a proper value measurement model [40,41] to assess the success of a project.

2.2. Multi-criteria evaluation

In a multi-criteria value model, the measure of success of a project is given by the additive value function model:

$$V(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} w_j v_j(x_j), \text{ with } \sum_{j=1}^{n} w_j = 1 \text{ and } w_j > 0, \forall_j \qquad (1)$$

Where V is the overall value score of the success of the project, w_j is the weight of criterion j, $v_j(x_j)$ is the value score on criterion j of the performance x_j, and $n$represents the number of evaluation criteria.

Despite being straightforward in form, this model is often poorly applied. We highlight that the criteria weights w_j are scaling constants [42], which represent trade-offs between criteria and not the erroneous notion of criteria's measures of importance [21]. In addition, v_j is a measurable value function, which represents both a preference order between performances on criterion j and a strength-of-preference order on differences of performances [43]. Moreover, the model requires the criteria to be mutually preferentially independent [44], which entails special care during the model structuring phase.

There are some fundamental aspects to note regarding the desired properties for each evaluation criterion and also for the whole set of criteria [45]. Each criterion should be *essential* for the evaluation and *controllable* in the sense that the performance of the project influences the degree to which the criterion is satisfied, independently of other additional decisions. Also, a family of evaluation criteria should be: *complete* (the set of criteria should represent all of the relevant consequences of the project); *nonredundant* (the criteria should not repeat the same concerns); *concise* (the number of criteria should be kept to the necessary minimum to evaluate the project); *specific* (each criterion should be able to assess the consequences of the project, instead of being so broad that it compromises this purpose); and *understandable* (the evaluation criteria should be clear in the eyes of any interested individual).

Depending on the ability to use appropriate numerical principles and fluency to express oneself in words, an evaluator may prefer to apply a numerical method or a non-numerical one [46]. In light of this, the remainder of this section focuses on quantitative and qualitative techniques tailored for these two types of evaluators. Specifically, we delve into methods for criteria weighting and building a value scale for each criterion.

2.2.1. Weighting methods

A theoretically sound weighting method must consider the performance ranges defined by two fixed references on each criterion. Common references are, for example, the "worst" and the "best" performances [39] or "neutral" and "good" performances [47]. Below, we briefly describe two quantitative weighting procedures and one qualitative.

Keeney and Raiffa [48] developed the *trade-off procedure*, which is a numerical method that requires establishing *indifferences* between two fictitious projects using two criteria at each time. After establishing $n - 1$ *indifference* relationships for the n criteria, a system of equations is solved, including one equation in which the sum of the weights equals 1, to obtain the criteria weights.

Edwards and Barron [49] created the *swing weighting method*, which is a numerical method that involves measuring the relative importance of the improvements (*swings*) that can be achieved on the criteria, considering a change from the "worst" to the "best" performance on each of them.

Bana e Costa and Vansnick [50] developed MACBETH [51] to weight the criteria. This procedure requires ranking the worst–best swings and judging them using the qualitative scale of difference in attractiveness: *no (difference), very weak, weak, moderate, strong, very strong*, or *extreme*. This qualitative scale is also used to judge the difference in attractiveness between two swings at a time. The elicited judgments are used to fill in the upper triangular part of a matrix in the software tool M-MACBETH, which validates each judgment's consistency with those previously inputted (see [52], pp. 425–443). Then, the software tool generates a proposal of weights compatible with the inputted qualitative judgments by solving the linear programming problem described in Bana e Costa et al. [52]. The evaluators should validate the proposed weighting scale and adjust it if needed.

2.2.2. Methods to build value scales

We must assign fixed scores to the previously defined references to build a criterion value scale. For example, we may assign 100 and 0 value units to the "best" and the "worst" performances in each criterion, respectively, although two other scores could be used so that the highest score is assigned to the most preferred reference. Though this arbitrary assignment of scores leads to obtaining *interval value scales* [25]. Additionally, the score of a project on a given criterion should consider the preferences expressed by the evaluators upon performance ranges within the criterion [43] (e.g., the difference in value between performances A and B is worth twice the difference between C and D). Hereinafter, we present two numerical scoring methods and a qualitative one.

Edwards [53] presents the *direct rating* method. This numerical procedure first requires evaluators to rank the project performances in order of decreasing attractiveness. The highest score (100 units) is assigned to the "best" performance and the lowest score (0 units) to the "worst". Intermediate scores are assigned to other performance levels considering the intensities of preferences between each two of them, knowing that the difference between the "best" and "worst" is worth 100 value units. This method allows scoring a project directly or indirectly using a performance measure (e.g., quantitative continuous, quantitative discrete, or qualitative). von Winterfeldt and Edwards [54] describe the *bisection* method, also known as the *mid-value splitting technique* [55], to create a value scale for a criterion. This numerical method assigns the highest score to the "best" performance (100) on the criterion and the lowest score (zero) to the "worst". Then, it is asked which performance p has a value equally distant from the "best" and the "worst" performances, which means that the ranges "p–to–best" and "p–to–worst" have the same strength-of-preference. Therefore, the performance p would get a midpoint score of 50. Similar midpoint questions are asked to identify other points that can be used to form a piecewise linear value function or a curve. This method allows the creation of value functions upon a quantitative and continuous performance measure on the criterion.

Bana e Costa and Vansnick [50] developed MACBETH [51] to create a value scale for a criterion (and to weight criteria, as described in the preceding section). Still, contrary to the above-mentioned methods, it needs only to elicit qualitative judgments. An evaluator judges the difference in attractiveness between two performances at a time, using the qualitative scale presented in the previous section, and inputs them into the software tool M-MACBETH. This tool verifies the consistency of the inputted judgments and generates a proposal of a value scale compatible with them and with the scores assigned to the reference performances "best" and "worst" (or "good" and "neutral") [52]. In the final step, the evaluator must validate and adjust the proposed value scale if needed. As in *direct rating*, this method allows scoring a project directly or indirectly using any performance measure.

2.3. Review summary

In the project success literature reviewed, most papers address the identification of IT criteria (e.g., Lobato et al. [4] and Assalaarachchi et al. [56]) or success factors (e.g., Pinheiro et al. [57] and Jayakody and Wijayanayake [58]), but only a few present an evaluation approach. In addition, the evaluation methods identified suffer from one or more theoretical errors (e.g., weights used as indicators of importance, averages calculated with ordinal scales, application of techniques with known flaws, and normalisation procedures that do not consider non-linear preferences). Furthermore, as far as we know, there is no description of a formal process that may guide the evaluators from beginning to end, i.e., from identifying the evaluation criteria until reaching an overall measure of project success. Therefore, a gap in the IT project literature needs to be addressed, which will be done by applying multi-criteria evaluation principles.

Given the characteristics of the evaluators, the simplicity of use of the MACBETH method and its software tool M-MACBETH, including its ability to validate the consistency of the value judgments expressed by evaluators and to work with any performance measure (be it qualitative or quantitative, continuous or discrete), this was the approach selected to weight the criteria and build a value function for each criterion in the real-world case described in this paper.

3. Model development

3.1. Research setting

GlobalSysMakers develops solutions in four business areas: mobility solutions, industrial technology, consumer goods, and energy and building technology. It has several divisions, including automobile multimedia, automobile accessories, electric tools, heating and hot water, and home appliances. It employs roughly 410,000 associates worldwide, has about 440 subsidiaries and regional companies in 60 countries, and employs nearly 70,000 associates in research and development at 125 locations.

The target project, here identified as PROJRD, was part of an R&D program that had the participation of GlobalSysMakers and a university. The project had as its primary goal the development of a software tool to automate the assessment of printed circuit boards (PCBs) design. PCBs are essentially boards that connect electronic components used in all (but the simplest) electronic products, such as household appliances or vehicles. In addition to the software tool, the project deliverables included technical specifications, prototypes, and presentations.

The software development process adopted was based on a hybrid/agile methodology supported by SCRUM [59]. Agile methods for software development have been increasingly used in the IT sector [60] and are now mainstream [61]. In this project, agility enabled greater adaptability of the development phases according to the company's needs and requirements, which evolved along with the project lifecycle. Thus, it was possible to deal with changes in the requirements that were reflected in the final deliverables during the project development. In a later phase of the project, the SCRUM was coupled with a waterfall process since the objectives stabilised without needing a periodic update. The project team was multidisciplinary, incorporating engineers from GlobalSysMakers (TEAMGSM) and researchers from the university (TEAMUNI). Together, the teams (TEAMGSM and TEAMUNI) had electronics, software engineering, and project management skills.

On average, the team allocated 1040 h per month to the project (approximately 6.5 Full-Time Equivalent), distributed by the different tasks of the project and according to the functions performed by each element (three of the team members were not full-time in the project). The project had a duration of 36 months.

The project's overall success was first assessed using a simple grid scoring model built by non-specialists in evaluation, which directly scored the project on several criteria and assigned importance weights. However, the project management team felt the need for a more advanced model to improve confidence in the evaluation. More in-depth research on multi-criteria evaluation revealed some misinterpretations in that process, which ultimately led to the development of a new model in line with decision analysis principles. This paper describes the new evaluation model.

3.2. Development tasks

The model development process started by asking the project manager to identify the members who should form the decision-making group [62], i.e., the group in charge of developing the model to evaluate the project's success. It was recommended to select members with different roles in the project; all of them were somehow interested in the project's outcomes. The group had three members: two from TEAMGSM and TEAMUNI, and one external consultant. The team members were selected considering their managerial responsibilities and to ensure representativeness of all the involved parties. All the members agreed to be involved in the model development tasks. Note that larger groups require different group processes, typically having separate meetings with stakeholders of different areas of interest to develop parts of the model, and with merge meetings gathering higher-level representatives of the client to validate the work done by the stakeholders and to finish the overall model [63].

Fig. 1 depicts the model development tasks. The first task involves identifying the aspects of interest for evaluating the project's success ("*problem structuring*", described in Section 3.3). This is a critical task because it is not possible to develop a proper evaluation model without understanding the problem, which is the reason why several publications have been devoted to identifying the fundamental evaluation concerns to be addressed (e.g., [28,64]). Second, all the relevant evaluation criteria should be included in the model, and a descriptor of performance should be identified for each of them, enabling the assessment of the extent to which each criterion is met ("*model structuring*", Section 3.4). Third, the evaluation component of the model must be built ("*value model building*", Section 3.5), which includes the construction of a value function for each criterion to transform the performances of the project into value scores (Section 3.5.1), and weighting the criteria to depict their trade-offs (Section 3.5.2). Last, the evaluation model should be tested for adequacy and consistency (Section 4.1).

3.3. Problem structuring

The problem structuring task aims to identify the fundamental objectives [45] that determine the project's success from the client's perspective. Such objectives are essential reasons for the project's success. Therefore, they should be used as criteria in the evaluation model.

However, the identification of these objectives in ill-structured problems may not be easy, which is why we opted to apply a *problem structuring method* (PSM) known as *group map* [65], which can be used in combination with a multi-criteria decision analysis approach [66].

To begin structuring the problem, the decision-making group was asked to say which aspects or concerns were relevant to evaluate the project's success. Then, for each of the concerns expressed, it was asked, "Why is that important?" or "What would be the consequences of doing that?", which allowed us to identify other aspects.

Fig. 2 depicts the complete group causal map built with the answers

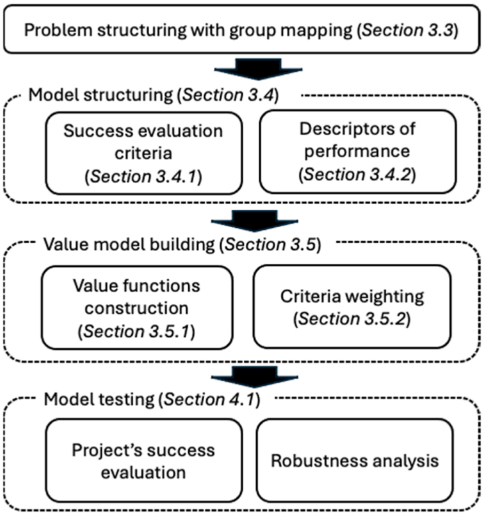

Problem structuring with group mapping (*Section 3.3*)
Model structuring (*Section 3.4*)
Success evaluation criteria (*Section 3.4.1*)
Descriptors of performance (*Section 3.4.2*)
Value model building (*Section 3.5*)
Value functions construction (*Section 3.5.1*)
Criteria weighting (*Section 3.5.2*)
Model testing (*Section 4.1*)
Project's success evaluation
Robustness analysis

Fig. 1. Model development tasks.

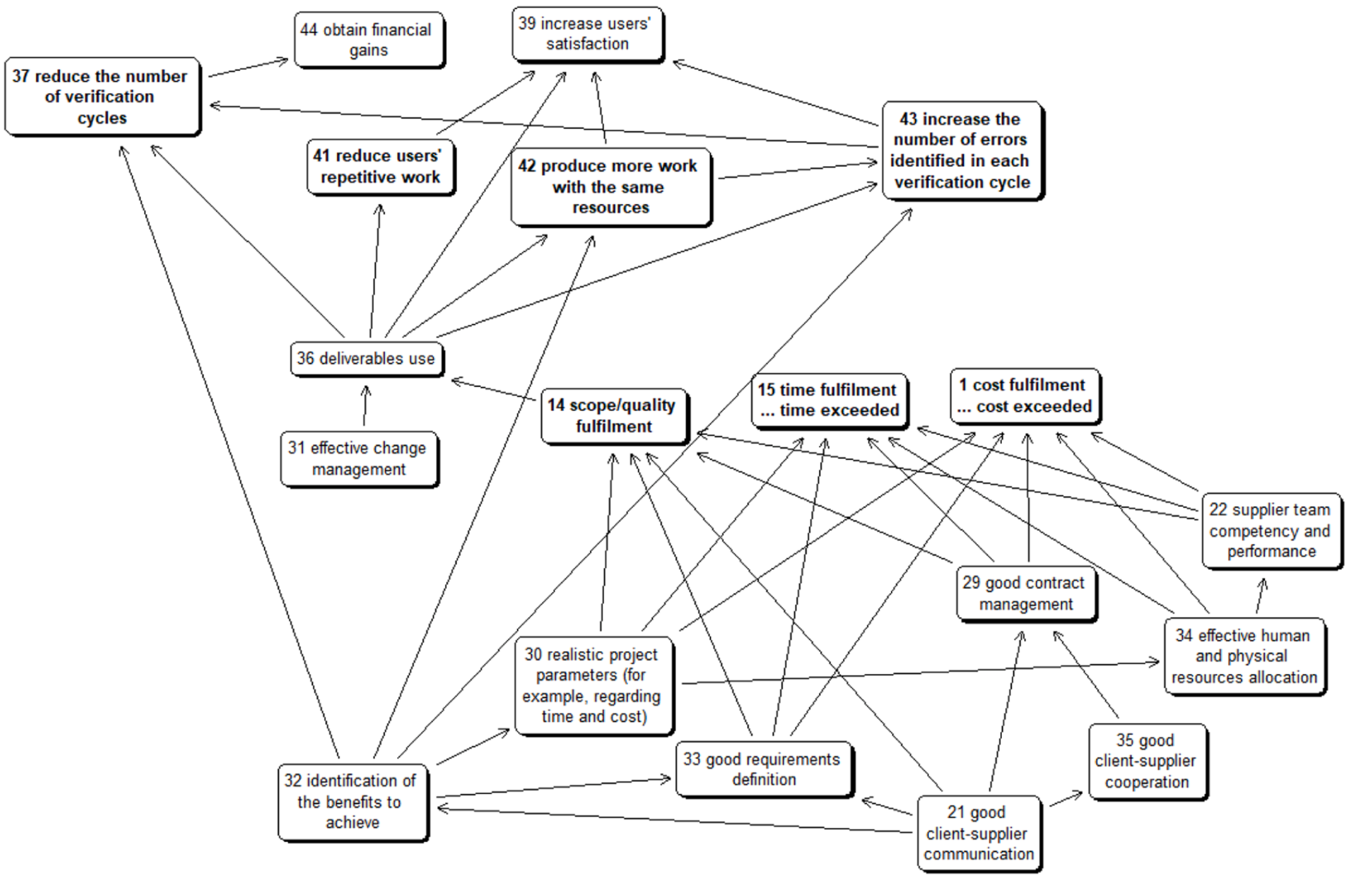

37 reduce the number of verification cycles
44 obtain financial gains
39 increase users' satisfaction
43 increase the number of errors identified in each verification cycle
41 reduce users' repetitive work
42 produce more work with the same resources
36 deliverables use
31 effective change management
14 scope/quality fulfilment
15 time fulfilment ... time exceeded
1 cost fulfilment ... cost exceeded
22 supplier team competency and performance
29 good contract management
34 effective human and physical resources allocation
30 realistic project parameters (for example, regarding time and cost)
35 good client-supplier cooperation
33 good requirements definition
32 identification of the benefits to achieve
21 good client-supplier communication

Fig. 2. Group map.

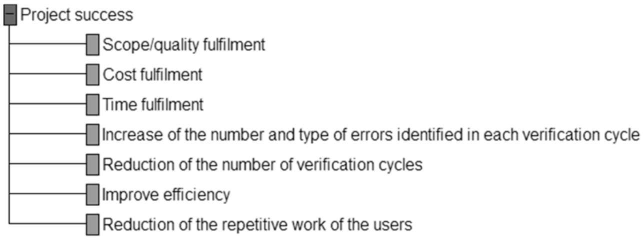
of the elements of the group using the software tool “Decision Explorer” (from Banxia Software Ltd., https://banxia.com/dexplore), which automatically numbered the concerns for identification purposes. This map results from several iterations, adding some aspects and removing others. Note that a specific *concern* may be expressed by one statement (e.g., “(33) good requirements definition”) or by two statements separated by an ellipsis, which depicts a positive pole and a negative one to clarify the meaning of the *concern* (e.g., “15 time fulfilment… time exceeded”). An arrow between two *concerns* indicates the direction of causality. When an arrow points to a *concern* with two poles, it means that the *concern* affected is the one at the positive pole (e.g., a “(29) good contract management” contributes to the positive pole of “(1) cost fulfilment… cost exceeded”; in the reverse case, the arrow would have a negative sign near its head).

In Fig. 2, it is possible to identify chains of means-ends objectives. For example, an “(31) effective change management” contributes to the “(36) deliverables use”, which respectively allows to “(41) reduce users' repetitive work”, which contributes to “increase users' satisfaction”. Although the “(41) reduce users' repetitive work” is a means-objective to the end-objective “(39) increase users' satisfaction”, the group considered the former a fundamental objective because it is important in itself and not because of its contribution to the latter. Therefore, “(41) reduce users' repetitive work” will be used as an evaluation criterion. Objective “(39) increase users' satisfaction” was considered too broad to evaluate the project’s success and thus will not be used.

3.4. Model structuring

3.4.1. Evaluation criteria

Fig. 3 depicts the seven evaluation criteria that emerged from the concerns highlighted in bold in the group causal map developed in the problem structuring task.



- Project success
  - Scope/quality fulfilment
  - Cost fulfilment
  - Time fulfilment
  - Increase of the number and type of errors identified in each verification cycle
  - Reduction of the number of verification cycles
  - Improve efficiency
  - Reduction of the repetitive work of the users

Fig. 3. Project’s success evaluation criteria.

The concerns represented by these criteria are as follows:

- *Scope/quality fulfilment (ScoQual)*—the extent to which the planned (functional and non-functional) requirements were fulfilled (this criterion resulted from concern 14 in Fig. 2).

The prime deliverable of the project is a software tool to support the PCB’s design assessment, the other deliverables being subsidiary to this tool. In the end, if the software tool does not comply with a minimum set of planned requirements, it will not be able to assess the PCB’s design and will compromise the investment objectives.

- *Cost fulfilment (Cost)*—the extent to which the planned cost was fulfilled (this criterion resulted from concern 1 in Fig. 2).

The budget defined for the project needs to be carefully managed due to being financed by an external R&D entity with a very narrow margin of deviation.

- *Time fulfilment (Time)*—the extent to which the planned time was fulfilled (this criterion resulted from concern 15 in Fig. 2).

Since this project is part of a large program, time fulfillment is a significant management aspect because all the program's projects must be finished simultaneously due to the program's constraints. In other words, not meeting the deadline in this project would mean completing it in whatever form it is in when the program reaches its end, complying or not with the scope, and delivering or not what was planned.

- *Increase of the number and type of errors identified in each verification cycle (IncNoType)*—the extent to which the number and type of errors identified in each PCB's verification cycle increase (this criterion resulted from concern 43 in Fig. 2).

Before the project was implemented in the company, the PCB designs had been checked mainly in a semi-automatic way by specialised engineers. Due to the many PCB components, details, and rules to review, it was virtually impossible to check all of the required features. The consequence was the late detection of some errors in more advanced stages of the projects, or, in other words, in later verification cycles. This accounts for the importance of the new software tool to increase the number and type of errors identified early on in each verification cycle, thereby reducing the design costs.

- *Reduction of the number of verification cycles (RNVC)*—the extent to which the number of verification cycles is reduced (this criterion resulted from concern 37 in Fig. 2).

A PCB typically needs to go through several verification cycles until it is free from errors and ready for production. When errors are detected in a verification cycle, the PCB design needs to be corrected and tested again, possibly requiring a new verification cycle. Each verification cycle of a PCB design implies high costs. Furthermore, there is the risk of detecting errors only at the production stage, with even more severe consequences. A primary expected result of the new software tool is to reduce the number of verification cycles by enabling the early detection of errors.

- *Improve efficiency (ImpEff)*—the extent to which the number of verified rules increases in each verification cycle without increasing the involved human resources (this criterion resulted from concern 42 in Fig. 2).

Since the process for verifying the PCB's design rules is semi-automatic, with a substantial part of manual labour, the current number of specialised engineers can only check some of the relevant aspects. With the new software tool, it is expected that the same number of engineers can check a greater number of design rules, not spending more time doing it.

- *Reduction of the repetitive work of the users (RRWU)*—the extent to which the number of rules manually verified is reduced in each verification cycle (this criterion resulted from concern 41 in Fig. 2).

In the semi-automatic verification of PCB's design rules, manual labour is repetitive and prone to errors due to the fatigue of specialists. Automating most of the rules' assessment is expected to reduce the repetitive work of these specialists and free them to perform other tasks.

3.4.2. *Descriptors of performance*

In this task, we associate a descriptor of performance with each evaluation criterion to measure how much the project satisfies the criterion. According to Keeney [45], a descriptor should be *unambiguous* (to describe the performances on the associated criterion clearly), *comprehensive* (to cover the range of possible performances on the criterion), *direct* (the descriptor levels should directly describe the performances on the corresponding criterion), *operational* (the information concerning the performances of the project can be obtained and value judgments can be made), *understandable* (performances and value judgments made using the descriptor can be clearly understood and communicated).

Table 1 presents the list of all the descriptors created to measure the performance of the project, as well as two reference performance levels, "neutral" and "good", for each of them. Note that the definition of two reference performance levels is required to weigh the criteria, allowing comparisons between criteria preference ranges and defining two fixed anchors for the value scales (see Section 2.2). Furthermore, the use of a "neutral" performance level (which corresponds to a performance that is neither positive nor negative on the criterion) and of a "good" performance level (which corresponds to a very positive performance on the criterion) allows to increase the understandability of the criterion, and are thus preferable to the "worst" and the "best" references used as examples in Section 2.2.

As shown in Table 1, the criteria *scope/quality fulfilment* and *increase in the number and type of errors identified in each verification cycle* do not have direct descriptors of performance. For these criteria, *constructed descriptors* were developed combining the characteristics inherent to those criteria, as explained next (Bana e Costa et al. [67] describe a detailed procedure for creating constructed descriptors).

To measure the performance of the project on the *scope/quality fulfilment* criterion, several requirements that deliver different contributions to the project's success were considered, following the MoSCoW method principles [68]. These requirements were classified into three types ("must have", "important to have", and "nice to have") and combined to obtain the performance levels of the descriptor presented in Table 2.

To measure the performance of the project on the *increase of the number and type of errors identified in each verification cycle* criterion, several combinations of the number and type of errors identified at each verification cycle (based on a past project) need to be considered (see Table 3). For example, a "5 % increase in the number of identified errors" and a "10 % increase in the type of identified errors" is a performance depicted as level "E5 T10". A verification cycle includes a series of tests to check for errors in the PCB's design or if it is ready for production (free from errors).

We note that the indicators used in the constructed scales presented in Tables 2 and 3 cannot be considered in isolation, as they are mutually preferentially dependent. For example, in Table 3, *an increase of 10 % in*

Table 1
Descriptors of performance.

Criterion	Descriptor	Neutral	Good
Scope/quality fulfilment *(ScoQual)*	Constructed descriptor (see Table 2)	L2	L3
Cost fulfilment *(Cost)*	Cost of the project (k€)	Planned cost (k€ 500)	95 % of the planned cost (k€ 450)
Time fulfilment *(Time)*	Project duration (weeks)	Planned time (96 weeks)	95 % of the planned time (90 weeks)
Increase in the number and type of errors identified in each verification cycle *(IncNoType)*	Constructed descriptor (see Table 3)	E5 T0	E10 T5
Reduction of the number of verification cycles *(RNVC)*	Number of verification cycles decreased	1 cycle	2 cycles
Improve efficiency *(ImpEff)*	Number of verified rules increased (%)	0 %	40 %
Reduction of the repetitive work of the users *(RRWU)*	Number of rules manually verified reduced (%)	0 %	10 %

Table 2
Scale for "scope/quality fulfilment" criterion.

Performance levels	
The project…	
…satisfied all the requirements "must have" and "important to have" and most of the "nice to have"	L1
…satisfied all the requirements "must have" and at least 85 % of the "important to have" and at least 20 % of the "nice to have" (or an equivalent performance on the requirements "important to have" and "nice to have")	L2 = Good
…satisfied all the requirements "must have" and at least 60 % of the "important to have" and at least 20 % of the "nice to have" (or an equivalent performance on the requirements "important to have" and "nice to have")	L3 = Neutral
…did not satisfy one requirement "must have", or satisfied less than 60 % of the requirements "important to have"	L4
…did not satisfy more than one requirement "must have"	L5

Table 3
Constructed scale for "increase of the number and type of errors identified in each verification cycle" criterion.

Increase in the number of identified errors (E)	Increase in the type of identified errors (T)	Level
10 %	10 %	E10 T10
10 %	5 %	E10 T5 = Good
10 %	0 %	E10 T0
5 %	10 %	E5 T10
5 %	5 %	E5 T5
5 %	0 %	E5 T0 = Neutral
0 %	0 %	E0 T0

the number of identified errors (E) is valued more highly when the *percentage increase in the type of identified errors (T)* is greater. Otherwise, the number and the type of identified errors could have been used as indicators for two separate evaluation criteria.

After the seven criteria had been clearly identified and their descriptors of performance established, the decision-making group was asked whether there was any additional aspect that might be considered in assessing the project's success. The negative response indicated that this set of criteria was exhaustive and, consequently, that the value tree presented in Fig. 3 could be considered complete.

3.5. Value model building

3.5.1. Value functions

As previously described, a descriptor of performance provides a way of measuring the project's performance on its associated criterion. However, to build a value model, we also need to obtain the value of each plausible performance of the project (in the form of a value scale or value function), which requires knowing the preferences of the evaluators upon differences in performances on the corresponding criterion.

For that purpose, we applied the MACBETH method [51]. As described in Section 2.2, the questioning procedure of MACBETH requires the evaluators to answer questions of difference in attractiveness between two performance levels at each time, using the qualitative scale: no (difference in attractiveness), very weak, weak, moderate, strong, very strong, and extreme. The answers provided are used for filling in a matrix of judgments in the M-MACBETH software tool, which analyses the consistency of the answers as soon as they are inserted, and then generates (by linear programming) a proposal of value scale which is compatible with the answers provided, given the fixed value scores assigned to the "neutral" and the "good" performances (0 and 100 value units, respectively).

We present two examples of applying the MACBETH method to build value functions for criteria with different descriptors of performance: *scope/quality fulfilment* criterion with a discrete descriptor, and *time fulfilment* criterion with a continuous descriptor.

Fig. 4 presents the matrix of judgments for the *scope/quality fulfilment* criterion. Table 2 shows the constructed descriptor for this criterion where: L1 means "the project satisfied all the requirements 'must have' and 'important to have' and the majority of the 'nice to have'", L2 means "the project satisfied all the requirements 'must have' and at least 85 % of the 'important to have' and at least 20 % of the 'nice to have' (or an equivalent performance)", and L3 means "the project satisfied all the requirements 'must have' and at least 60 % of the 'important to have' and at least 20 % of the 'nice to have' (or an equivalent performance)". We can see in Fig. 4 that the difference in attractiveness between "L1" and "L2 = Good" was deemed *weak* by the evaluators, whereas the difference in attractiveness between "L2 = Good" and "L3 = Neutral" was considered *moderate*. Therefore, the difference in value between "L1" and "L2 = Good" should be lower than the difference between "L2 = Good" and "L3 = Neutral", which can be confirmed in the value scale presented in Fig. 6a, where the former difference corresponds to 65 value units and the latter to 100.

The *time fulfilment* criterion has the descriptor of performance "project duration (in weeks)" with the references "96 weeks = Neutral" and "90 weeks = Good". To build a value function for this criterion, first, we created three more equally spaced performance levels: one worse than "neutral" (99 weeks), one between "neutral" and "good" (93 weeks), and one better than "good" (87 weeks). Then, the evaluators judged the differences in attractiveness between each two of these levels, together with the "neutral" and the "good" levels, resulting in the matrix of judgments presented in Fig. 5.

Looking at the diagonal (above the grey shaded cells) of the matrix in Fig. 5 we see that the intensities of the differences in attractiveness between each two consecutive levels increase more when the number of weeks exceeds 93 weeks: the evaluators considered *weak* the differences in attractiveness between "87" and "90 = Good" (and also between "90 = Good" and "93"), whereas they considered *moderate* the difference in attractiveness between "93" and "96 = Neutral", and *very strong* the difference between "96 = Neutral" and "99". Therefore, the difference in value between "87" and "90 = Good" (and also between "90 = Good" and "93") should be lower than the difference in value between "93" and "96 = Neutral", and the latter should also be lower than the difference in value between "96 = Neutral" and "99", which can be confirmed in the value function presented in Fig. 6c (each of the first two intervals corresponds to 40 value units, whereas the third and fourth equal 60 value units and 160, respectively). Therefore, this function shows that the evaluators considered that increments in time after 93 weeks are increasingly penalizing for the project's success.

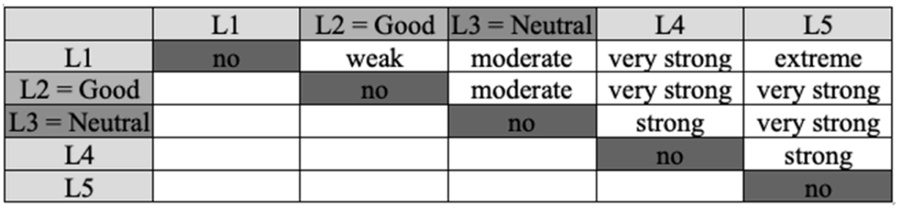
We emphasize that the decision group made these judgments for each criterion independently of the performance levels or the differences in attractiveness on the remaining criteria, thereby supporting the assumption of mutual preferential independence between criteria.

Fig. 6 (6a–6g) presents the value functions of all the evaluation criteria.

3.5.2. Criteria weighting

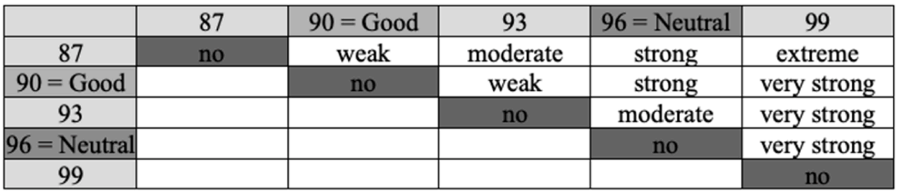
Weighting requires establishing trade-offs between criteria, which is typically demanding because it implies comparing performance improvements on different criteria. The improvements (*swings*) are defined between the two predefined performance references, "neutral" and "good", in each criterion.

According to the MACBETH weighting procedure, the first step was to rank the "neutral–good" swings in order of decreasing preference (Fig. 7). The evaluators considered the swing from "1 to 2 verification cycles decreased" as the most important one (1st in Fig. 7), which implies that the criterion "reduction of the number of verification cycles (RNVC)" will have the highest weight. In contrast, the criterion "reduction of repetitive work of the users (RRWU)" will obtain the lowest weight because it has the least important "neutral–good" swing



	L1	L2 = Good	L3 = Neutral	L4	L5
L1	no	weak	moderate	very strong	extreme
L2 = Good		no	moderate	very strong	very strong
L3 = Neutral			no	strong	very strong
L4				no	strong
L5					no

Fig. 4. MACBETH judgment matrix for the “Scope/quality fulfilment” criterion.



	87	90 = Good	93	96 = Neutral	99
87	no	weak	moderate	strong	extreme
90 = Good		no	weak	strong	very strong
93			no	moderate	very strong
96 = Neutral				no	very strong
99					no

Fig. 5. MACBETH judgment matrix for the “time fulfilment” criterion.

(7th in Fig. 7).

In the second step, the improvements provided by the criteria swings were judged qualitatively using the MACBETH semantic scale (Fig. 8), which allowed filling in the rightmost column in Fig. 9. For example, the improvement provided by the most important swing [RNVC] was considered *extreme*, whereas the least important “neutral–good” swing [RRWU] was judged *weak*.

Then, the differences in attractiveness between each two “neutral–good” swings were assessed to fill in the remaining cells of the first row of the weighting matrix and fill in the diagonal above the shaded cells in Fig. 9. For example, Fig. 10 depicts the comparison of the “neutral–good” swings in the *reduction of the number of verification cycles* (RNVC) criterion and in the *increase in the number and type of errors identified in each verification cycle* (IncNoType) criterion, which was deemed as *very strong* (*v. strong* in Fig. 9). The other cells with no judgments were filled in automatically (by transitiveness) with “P” (positive) judgments by M-MACBETH.

Finally, the software tool applied the linear programming model described in Bana e Costa et al. [51] to generate a proposal of a weighting scale consistent with the qualitative judgments expressed in the weighting matrix, which were subsequently validated by the evaluators (with some minor adjustments), resulting in the weights presented in Fig. 11.

4. Results and discussion

4.1. Model testing and results

At this point, the actual performances of the project are already known for most of the criteria, but not for the *reduction of the number of verification cycles (RNVC)* criterion, which will only be identified in the long term. Therefore, three alternative scenarios were created with hypothetical future performances on *RNCV*: no reduction at all (PCB no red cycles), a decrease of one verification cycle (PCB red 1 cycle), and a decrease of two verification cycles (PCB red 2 cycles). The performances of these scenarios are shown in Table 4.

Applying the value functions previously defined for each criterion to the performances presented in Table 4, we obtain the partial and the overall value scores of the three scenarios shown in Table 5 using the previously assessed criteria weights.

As seen in Table 5, the most advantageous scenario corresponds to “PCB red 2 cycles” with 94.60 overall value units, followed by “PCB red 1 cycle” with 49.60, and “PCB no red of cycles” with –6.65.

Scenarios “PCB red 2 cycles” and “PCB red 1 cycle” undoubtedly denote a successful project independently of the weights assigned to criteria, because their performances are not worse than “neutral” in any of the criteria and are better than it in several criteria. Therefore, both scenarios *dominate* [69] a “neutral project”. Additionally, we may see that scenario “PCB red 2 cycles” has an overall score very close to that of a “good project” (100 units), whereas the value of scenario “PCB red 1 cycle” is almost mid-distance from a “neutral project” and a “good project”.

However, it is not robust to say that the scenario “PCB no red of cycles” corresponds to an unsuccessful project, looking only at its overall value score. We must determine if its overall result will always be worse than that of a “neutral project” when in the face of the uncertainty defined for the model parameters (i.e., the value scores and criteria weights). In fact, the evaluators considered it plausible that: a) each criterion weight ($w_j, j = 1, \ldots, 7$) may vary within an interval defined by the lower and upper limits $\left(\underline{w}_j \leq w_j \leq \ \overline{w}_j,\ j = 1, \ldots, 7\right)$ shown in Table 6; and b) the value scores of the scenario “PCB no red of cycles” may have plus or minus 5 value units (respectively denoted by $\overline{v}_j\left(y_j\right)$ and $\underline{v}_j\left(y_j\right)$, $j\ = 1, \ldots, 7)$ in all the criteria for which this scenario has a performance different from “neutral” and “good”, otherwise it will keep 0 and 100, respectively.

The linear programming (LP) problem (2) was then used to test whether a “neutral project” *additively dominates* [70] the scenario “PCB no red of cycles”, which would require a negative max*D*. The result max*D* = 9.575denotes that there is at least one combination of plausible scores and weights for which scenario “PCB no red of cycles” has a higher overall value than that of a “neutral project”.

The worst possible overall value for scenario “PCB no red of cycles” was also calculated, with the LP problem (3), resulting in min*D* = –14.10. Therefore, in the face of the uncertainty, the overall value score of scenario “PCB no red of cycles” may vary between –14.10 and 9.575.

$$\max D = \sum_{j=1}^{7} w_j \left[\overline{v}_j\left(y_j\right) - v_j\left(\text{neutral}_j\right)\right] \tag{2}$$

Subject to:

$$\sum_{j=1}^{7} w_j = 1$$

$$\underline{w}_j \leq w_j \leq \ \overline{w}_j,\ j = 1, \ldots, 7$$

$$\min D = \sum_{j=1}^{7} w_j \left[\underline{v}_j\left(y_j\right) - v_j\left(\text{neutral}_j\right)\right] \tag{3}$$

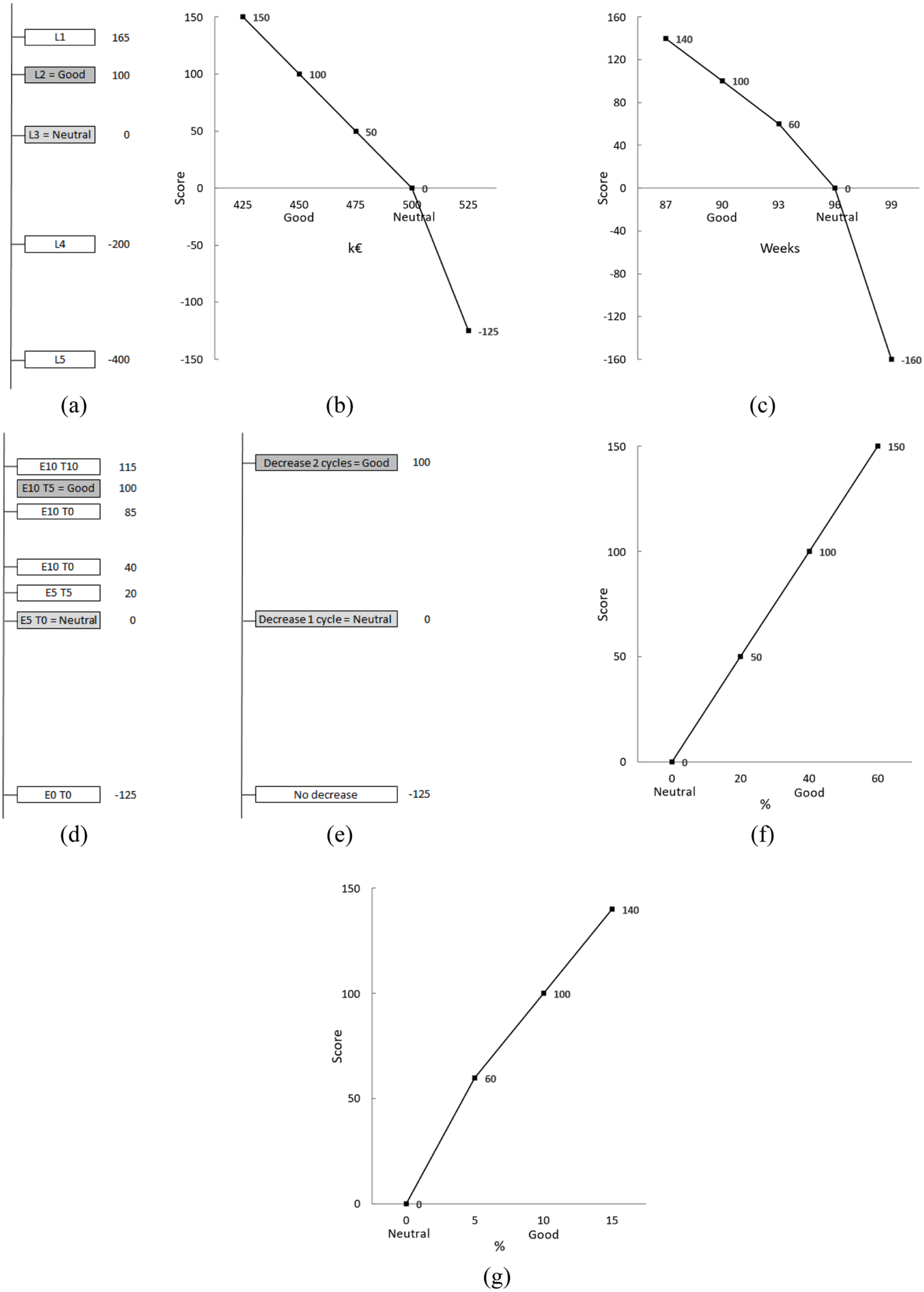


Fig. 6. Value functions of criteria: (a) scope/quality fulfilment, (b) cost fulfilment, (c) time fulfilment, (d) increase in the number and type of errors identified in each verification cycle, (e) reduction of the number of verification cycles, (f) improve efficiency, (g) reduction of the repetitive work of the users.

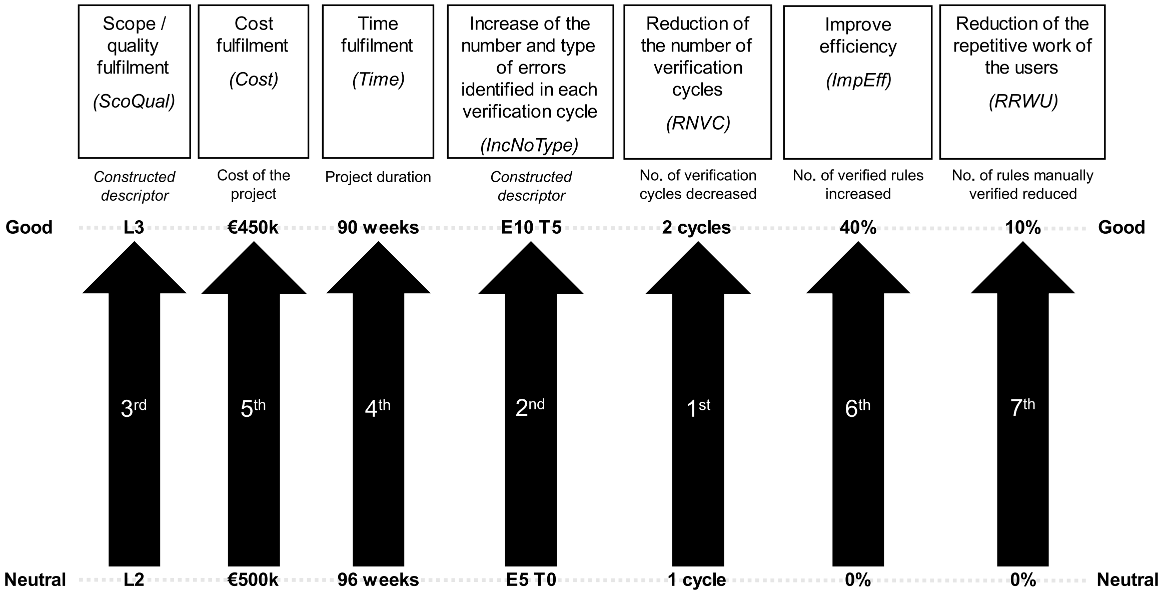

Scope / quality fulfilment (ScoQual)
Cost fulfilment (Cost)
Time fulfilment (Time)
Increase of the number and type of errors identified in each verification cycle (IncNoType)
Reduction of the number of verification cycles (RNVC)
Improve efficiency (ImpEff)
Reduction of the repetitive work of the users (RRWU)
Constructed descriptor
Cost of the project
Project duration
Constructed descriptor
No. of verification cycles decreased
No. of verified rules increased
No. of rules manually verified reduced
Good
L3
€450k
90 weeks
E10 T5
2 cycles
40%
10%
Good
3rd
5th
4th
2nd
1st
6th
7th
Neutral
L2
€500k
96 weeks
E5 T0
1 cycle
0%
0%
Neutral

Fig. 7. Neutral–good swings ranking.

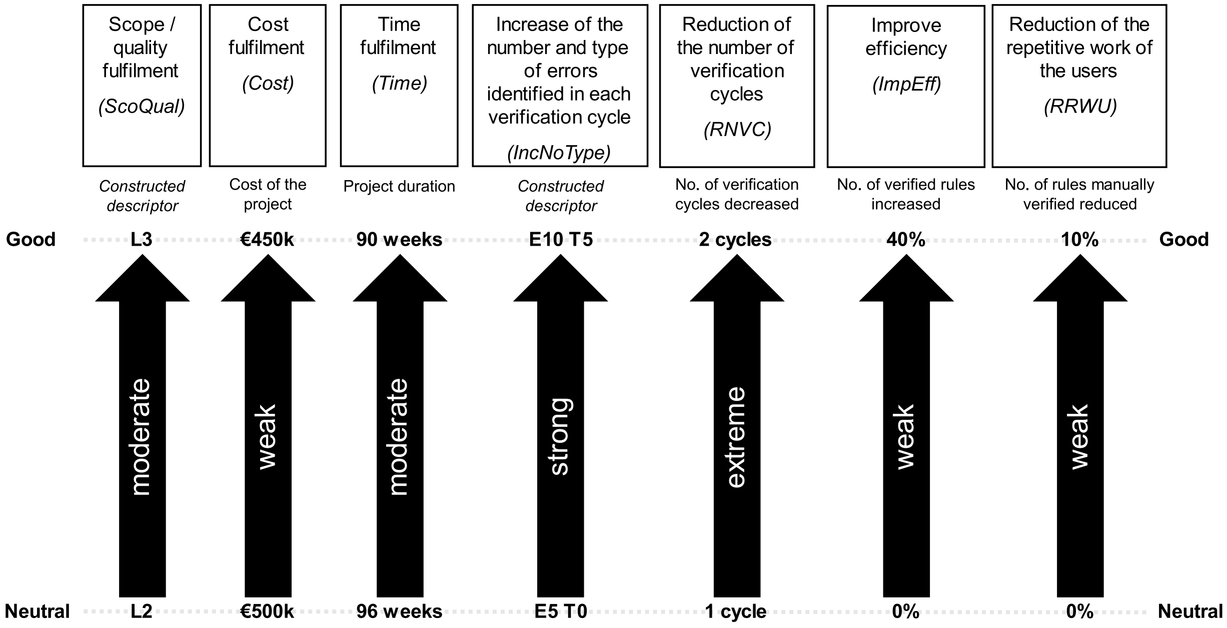
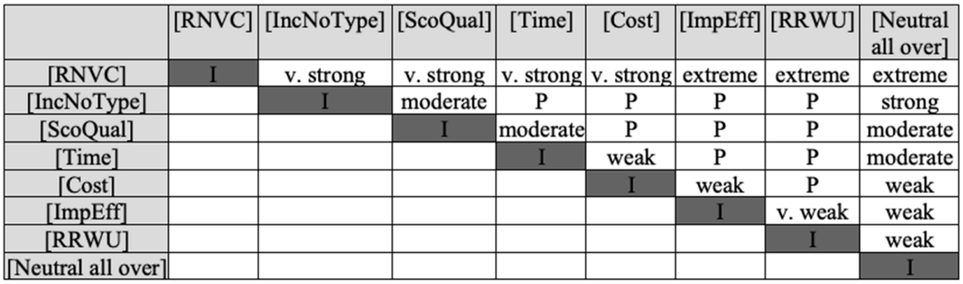

Scope / quality fulfilment (ScoQual)
Cost fulfilment (Cost)
Time fulfilment (Time)
Increase of the number and type of errors identified in each verification cycle (IncNoType)
Reduction of the number of verification cycles (RNVC)
Improve efficiency (ImpEff)
Reduction of the repetitive work of the users (RRWU)
Constructed descriptor
Cost of the project
Project duration
Constructed descriptor
No. of verification cycles decreased
No. of verified rules increased
No. of rules manually verified reduced
Good
L3
€450k
90 weeks
E10 T5
2 cycles
40%
10%
Good
moderate
weak
moderate
strong
extreme
weak
weak
Neutral
L2
€500k
96 weeks
E5 T0
1 cycle
0%
0%
Neutral

Fig. 8. Neutral–good swings' weighting judgments.



	[RNVC]	[IncNoType]	[ScoQual]	[Time]	[Cost]	[ImpEff]	[RRWU]	[Neutral all over]
[RNVC]	I	v. strong	v. strong	v. strong	v. strong	extreme	extreme	extreme
[IncNoType]		I	moderate	P	P	P	P	strong
[ScoQual]			I	moderate	P	P	P	moderate
[Time]				I	weak	P	P	moderate
[Cost]					I	weak	P	weak
[ImpEff]						I	v. weak	weak
[RRWU]							I	weak
[Neutral all over]								I

Fig. 9. MACBETH weighting matrix (the P and I within the matrix respectively mean positive difference in attractiveness and indifference).

subject to:

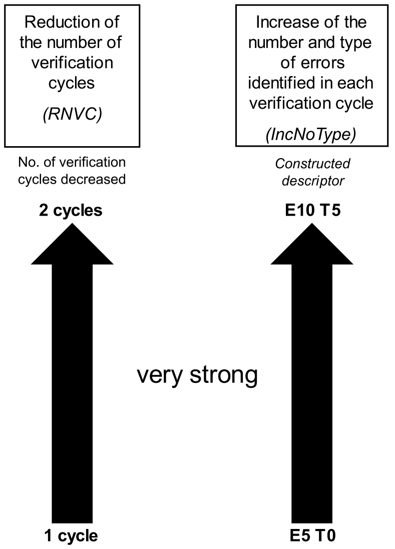

Reduction of the number of verification cycles
(RNVC)
Increase of the number and type of errors identified in each verification cycle
(IncNoType)
No. of verification cycles decreased
Constructed descriptor
2 cycles
E10 T5
very strong
1 cycle
E5 T0

Fig. 10. Assessment of the difference in attractiveness between the "neutral–good" swings in RNVC and IncNoType.

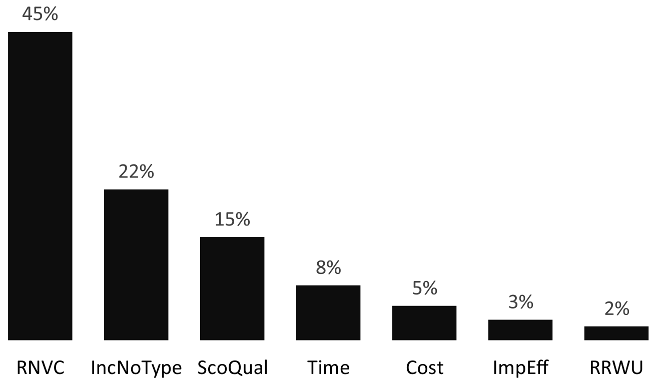


Fig. 11. Criteria weights.

$$\sum_{j=1}^{7} w_j = 1$$

$$\underline{w}_j \leq w_j \leq \overline{w}_j, \ j = 1, \ldots, 7$$

After concluding the robustness analysis, the evaluation group revisited the model and considered that it could deal with all the plausible performances and adequately considered the value judgments of its members. Therefore, the model has a form and content sufficient to evaluate the project's success [71].

5. Discussion

The absence of a formal evaluation of project success results in the waste of relevant lessons that can be used to enhance project management practices [9,72]. This is a strong reason for implementing well-structured processes to evaluate project success.

Any evaluation process should start by identifying the success criteria according to the decision-makers' preferences and systems of values, which are inherently subjective. We underscore that an evaluation model has an objective component (factual data) and a subjective one (value judgments), which should be independently addressed. Therefore, subjectivity is a key component in an evaluation process, but it should not be confused with ambiguity, which should be avoided. That is why the success evaluation criteria should be carefully identified, and a measure of the performance of a project on each of those criteria must be operationalised. The "neutral" and "good" references of intrinsic value allow identifying the project's success level.

Throughout the development of the evaluation model, the members of the decision-making group were encouraged to engage in open discussion whenever differences of opinion arose. This approach enabled a better understanding of their points of view and helped the group reach an agreement on the way forward.

In the case described herein, the success of the project may depend on the future performance of the *reduction of the number of verification cycles (RNVC)* criterion. With "no reduction of verification cycles", the project may be unsuccessful, with –6.65 overall value units, caused by its low performance and corresponding negative score (–125 value units) on this criterion. However, as we have seen, given the uncertainty defined for the partial value scores and the criteria weights, this scenario is not guaranteed to correspond to a negative evaluation. In fact, its overall value may vary between –14.10 and 9.575 units.

With a "reduction of 1 verification cycle", the project would obtain 49.60 overall value units, which is nearly a mid-distance evaluation between a "good project" and a "neutral project". With a "reduction of 2 verification cycles", the project would obtain 94.60 overall value units, which is very close to that of a "good project".

Developing a transparent evaluation process, such as the one described here, will promote the decision-making group's understanding and acceptance of the results. The participation of the decision-makers in all of the process phases is a key element for this purpose, which will allow them to develop a sense of ownership of the model [63]. However, this is not a practice found in the literature related to evaluating project success, which offers an opportunity for improvement.

The proposed process, which integrates a problem structuring

Table 4
Performance profiles of the project's success for the three scenarios.

Scenario / Criterion	ScoQual	Cost (k€)	Time (weeks)	IncNoType	RNVC	ImpEff (%)	RRWU (%)
PCB no red of cycles	L2	480	96	E10 T10	No decrease	60	15
PCB red 1 cycle	L2	480	96	E10 T10	Decrease 1 cycle	60	15
PCB red 2 cycles	L2	480	96	E10 T10	Decrease 2 cycles	60	15

Table 5
Value scores of the project success for the three scenarios.

Scenario / Criterion	ScoQual (15 %)	Cost (5 %)	Time (8 %)	IncNoType (22 %)	RNVC (45 %)	ImpEff (3 %)	RRWU (2 %)	Overall value score
PCB no red of cycles	100	40	0	115	–125	150	140	–6.65
PCB red 1 cycle	100	40	0	115	0	150	140	49.60
PCB red 2 cycles	100	40	0	115	100	150	140	94.60

Table 6
Plausible intervals for the criteria weights.

Criterion	ScoQual	Cost	Time	IncNoType	RNVC	ImpEff	RRWU
Index (j)	1	2	3	4	5	6	7
Current weight (w_j)	15 %	5 %	8 %	22 %	45 %	3 %	2 %
Upper limit ($\overline{w}_j$)	18 %	7 %	10 %	25 %	45 %	4 %	2.5 %
Lower limit ($\underline{w}_j$)	12 %	5 %	8 %	19 %	40 %	3 %	2 %

method with a multi-criteria decision analysis (MCDA) approach for evaluating the success of information technology (IT) projects, offers several significant theoretical contributions to the fields of project management, decision sciences, and IS. First, it advances the conceptual understanding of IT project success by addressing its inherently multidimensional and context-dependent nature. Traditional models often rely on narrow success criteria—such as time, cost, and scope—while this research introduces a more holistic and stakeholder-sensitive framework. By incorporating problem structuring methods, the process facilitates the elicitation and organization of the stakeholder perspectives, which are often overlooked or underrepresented in conventional evaluation models. This contributes to theory by emphasizing the social and interpretive dimensions of project success, aligning with contemporary views that success is not an objective outcome but a negotiated construct [73].

Second, the integration of MCDA techniques provides a rigorous and transparent mechanism for prioritizing and aggregating evaluation criteria, thereby enhancing the methodological robustness of success assessment. This methodological synthesis bridges a gap in the literature by demonstrating how qualitative insights from problem structuring can be systematically translated into quantitative decision models. Theoretically, this supports the development of hybrid evaluation frameworks that are both contextually grounded and analytically sound. Third, the application of the proposed process in a real-world case adds empirical depth to the theoretical model, offering evidence of its practical relevance and adaptability. This empirical grounding strengthens the external validity of the framework and encourages further theoretical exploration across different organizational and project contexts.

The MACBETH approach has been successfully employed, with different nuances and across various processes, to evaluate projects or decision alternatives in diverse problem settings and for a wide range of organizations [74]. The process described in this paper, which combines problem structuring with the MACBETH approach and robustness analysis, may also be applied in other contexts, subject to the necessary adjustments.

Our proposed process can also be scaled to the program or portfolio level, although this should be done with caution. In the case presented here, we applied an additive value function model, which is compensatory—meaning that poor performance on one criterion can be offset by good performance on others. However, this assumption may not always hold. In a program or portfolio context, for instance, if a key project performs poorly, that alone may render the entire program or portfolio unsuccessful, regardless of the performance of the remaining projects. In such cases, a mixed model should be adopted, combining classification rules to address the non-compensatory criteria with an additive component for the compensatory ones.

Moreover, the research highlights the absence of standardized approaches for evaluating IT project success, which has long been a limitation in both academic and professional domains. Standardization facilitates the dissemination of knowledge and enhances predictability, thereby minimizing uncertainty and reducing risk [75]. By proposing a replicable and adaptable process, the study lays the groundwork for the development of formalized evaluation standards. This has implications for theory-building, as it suggests a pathway toward unifying fragmented evaluation practices under a coherent, theoretically informed model. In doing so, it contributes to the ongoing discourse on standardization in project management and information systems evaluation, encouraging future research to refine, validate, and extend the proposed framework. Ultimately, this work not only enriches theoretical understanding but also provides a foundation for more consistent, transparent, and stakeholder-aligned evaluation practices in the IT project domain.

6. Conclusions

Evaluating the success of IT projects should be a mandatory project management activity. However, this is not observed in the practice [11, 72]. There are several contributions given by the process herein described, which can be easily adapted to other evaluation problems:

- It shows how a multi-criteria approach may be used to evaluate IT (software development) projects while avoiding committing critical mistakes.
- It offers a transparent process.
- It involves the decision-makers in all of the model development tasks.
- It identifies the fundamental objectives of decision-makers with the help of a problem structuring method, avoiding ending up solving the wrong problem [76].
- It allows establishing quantitative and substantive meaningful [23] trade-offs between criteria (i.e., mathematically valid and unambiguously understood).
- It allows the management of the project to focus on what matters for the project's success.
- It can be implemented to evaluate the success of other projects, in similar or different contexts.
- The use of descriptors of performance clarifies what is intended to be achieved in each criterion.
- It distinguishes performance from value, instead of directly attributing scores to the project, mixing these two components.
- And, it allows creating value scales adjusted to the preferences of evaluators, upon different types of performance (e.g., qualitative or quantitative, continuous or discrete).

Additionally, it enables the identification of alternative scenarios to deal with unknown future performances and to test the robustness of the conclusions considering uncertainties on the model parameters.

In the target organization, given the shortcomings recognised in a previous "grid scoring model", the multi-criteria evaluation model of the real-world case described in this paper was built during an advanced stage of the project's development. This late development can be considered a threat to internal validity regarding consistency and a limitation since the evaluation model should be built during the planning phase of a project and revisited during the project development to be improved, if needed, or adjusted to possible changes to the project aim. Another threat to external validity should also be disclosed. Namely, concerning scalability, further research is needed to test if the proposed process can be scaled or adapted for different project sizes or types.

In future work, it would be interesting to create a process capable of dealing with all project phases, allowing the evaluation of its development and evolution at several milestones, from the project initiation until its termination. The process described in this paper may be extended to evaluate project success throughout the project lifecycle. This requires developing a model that includes both final and intermediate objectives (criteria) for measuring project success. The intermediate objectives should be used during project development and later deactivated by setting their weights to zero and rescaling the remaining criteria weights so that they sum to one. Monitoring the evolution of a project's success against a well-defined set of criteria will allow identifying problems sooner and taking proper measures in time. Furthermore, the integration of the proposed evaluation process in the success management process [77] will add value to the management efforts.

Finally, since artificial intelligence technology, especially with the rise of Large Language Models (LLMs), has shown great potential in revolutionizing the automation of various complex tasks [78], it is imperative to explore it in the context of success evaluation.

CRediT authorship contribution statement

João Carlos Lourenço: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **João Varajão:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Unit Project Scope UID/00319/2025 - Centro ALGORITMI (ALGORITMI/UM). João C. Lourenço acknowledges the financial support of Portuguese funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the project UID/97/2025 (CEGIST). João C. Lourenço acknowledges the financial support of Portuguese funds through FCT – Fundação para a Ciência e a Tecnologia, I.P., under the project UID/97/2025 (CEGIST).

Data availability

The data is presented in the article.

References

[1] R. Colomo-Palacios, I. González-Carrasco, J.L. López-Cuadrado, A. Trigo, J. E. Varajao, I-Competere: using applied intelligence in search of competency gaps in software project managers, Inf. Syst. Front. 16 (4) (2014) 607–625, https://doi.org/10.1007/s10796-012-9369-6.

[2] M.A. Kafaji, Interchange roles of formal and informal project management on business operational success, Prod. Plan. Control (2022) 1–21, https://doi.org/10.1080/09537287.2022.2089265.

[3] L.A. Ika, J.K. Pinto, The "re-meaning" of project success: updating and recalibrating for a modern project management, Int. J. Proj. Manag. 40 (7) (2022) 835–848, https://doi.org/10.1016/j.ijproman.2022.08.001.

[4] B. Lobato, J. Varajão, C. Tam, A.A. Baptista, CrEISPS–a framework of criteria for evaluating success in information systems projects, Procedia Comput. Sci. 256 (2025) (2025) 1821–1835, https://doi.org/10.1016/j.procs.2025.02.323.

[5] N. Agarwal, U. Rathod, Defining 'success' for software projects: an exploratory revelation, Int. J. Proj. Manag. 24 (4) (2006) 358–370, https://doi.org/10.1016/j.ijproman.2005.11.009.

[6] R. Atkinson, Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria, Int. J. Proj. Manag. 17 (6) (1999) 337–342, https://doi.org/10.1016/S0263-7863(98)00069-6.

[7] H. Landrum, V.R. Prybutok, X. Zhang, The moderating effect of occupation on the perception of information services quality and success, Comput. Ind. Eng. 58 (1) (2010) 133–142, https://doi.org/10.1016/j.cie.2009.09.006.

[8] J.K. Pinto, D.P. Slevin, Project success: definitions and measurement techniques, Proj. Manag. J. 19 (1) (1988) 67–72.

[9] J. Varajão, L. Magalhães, L. Freitas, P. Rocha, Success management–from theory to practice, Int. J. Proj. Manag. 40 (5) (2022) 481–498, https://doi.org/10.1016/j.ijproman.2022.04.002.

[10] J. Varajão, J.C. Lourenço, J. Gomes, Models and methods for information systems project success evaluation–a review and directions for research, Heliyon 8 (12) (2022), https://doi.org/10.1016/j.heliyon.2022.e11977.

[11] J. Varajão, J.Á. Carvalho, Evaluating the success of IS/IT projects: how are companies doing it?, in: Proceedings of the 13th Pre-ICIS International Research Workshop on IT Project Management (IRWITPM 2018), San Francisco, USA, 2018.

[12] R.L. Keeney, Common mistakes in making value trade-offs, Oper. Res. 50 (6) (2002) 935–945, https://doi.org/10.1287/opre.50.6.935.357.

[13] J.E. Russo, P.J.H. Schoemaker, Decision Traps: The Ten Barriers to Brilliant Decision-Making and How to Overcome Them, Doubleday, 1989.

[14] S. Lipovetsky, A. Tishler, D. Dvir, A. Shenhar, The relative importance of project success dimensions, R&D Manag. 27 (2) (1997) 97–106, https://doi.org/10.1111/1467-9310.00047.

[15] Shapiro, J. (2005). Monitoring and evaluation. C.-W. A. f. C. Participation. https://www.civicus.org/view/media/Monitoring%20and%20Evaluation.pdf.

[16] Kahan, B., & Goodstadt, M. (2005). The IDM manual: basics. http://sites.utoronto.ca/chp/download/IDMmanual/IDM_basics_dist05.pdf.

[17] V. Arumugam, J. Antony, M. Kumar, Linking learning and knowledge creation to project success in Six Sigma projects: an empirical investigation, Int. J. Prod. Econ. 141 (1) (2013) 388–402, https://doi.org/10.1016/j.ijpe.2012.09.003.

[18] R. Linzalone, G. Schiuma, A review of program and project evaluation models, Meas. Bus. Excell. 19 (3) (2015) 90–99, https://doi.org/10.1108/MBE-04-2015-0024.

[19] P.L. Bannerman, A. Thorogood, Celebrating IT projects success: a multi-domain analysis, in: Proceedings of the 45th Hawaii International Conference on System Sciences, Maui, HI, 2012.

[20] C. Barclay, K. Osei-Bryson, Determining the contribution of IS projects: an approach to measure performance, in: Proceedings of the 42nd Hawaii International Conference on System Sciences, Waikoloa, HI, 2009.

[21] R.L. Keeney, Value-Focused Thinking: A Path to Creative Decisionmaking, Harvard University Press, 1992.

[22] R. Solingen, E. Berghout, The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development, McGraw-Hill, 1999.

[23] S. French, Decision Theory: An Introduction to the Mathematics of Rationality, Ellis Horwood, 1986.

[24] R. Göb, C. McCollin, M. Ramalhoto, Ordinal methodology in the analysis of Likert scales, Qual. Quant. 41 (5) (2007) 601–626, https://doi.org/10.1007/s11135-007-9089-z.

[25] S.S. Stevens, On the theory of scales of measurement, Science 103 (2684) (1946) 677–680, https://doi.org/10.1126/science.103.2684.677.

[26] W. Edwards, J.R. Newman, Multiattribute evaluation, in: T. Connolly, H.R. Arkes, K.R. Hammond (Eds.), Judgment and Decision Making: An Interdisciplinary Reader, 2nd ed, Cambridge University Press, 2000, pp. 17–34.

[27] R. von Nitzsch, M. Weber, The effect of attribute ranges on weights in multiattribute utility measurements, Manag. Sci. 39 (8) (1993) 937–943, https://doi.org/10.1287/mnsc.39.8.937.

[28] A. Basar, A novel methodology for performance evaluation of IT projects in a fuzzy environment: a case study, Soft Comput. 24 (14) (2020) 10755–10770, https://doi.org/10.1007/s00500-019-04579-y.

[29] H.N. Ismail, Measuring success of water reservoir project by using delphi and priority evaluation method, in: Proceedings of the IOP Conference Series: Earth and Environmental Science 588, 2020 042021, https://doi.org/10.1088/1755-1315/588/4/042021.

[30] J.H. Yu, H.R. Kwon, Critical success factors for urban regeneration projects in Korea, Int. J. Proj. Manag. 29 (7) (2011) 889–899, https://doi.org/10.1016/j.ijproman.2010.09.001.

[31] A. Nguvulu, S. Yamato, T. Honma, Project performance evaluation using deep belief networks, IEEJ Trans. Electron. Inf. Syst. 132 (2) (2012) 306–312, https://doi.org/10.1541/ieejeiss.132.306.

[32] C. Wohlin, A.A. Andrews, Assessing project success using subjective evaluation factors, Softw. Qual. J. 9 (1) (2001) 43–70, https://doi.org/10.1023/a:1016673203332.

[33] X. Yan, Utilizing the BSC method for IT performance evaluation of construction companies, in: Proceedings of the First International Conference on Information Science and Engineering, Nanjing, China, 2009.

[34] R.S. Kaplan, D.P. Norton, The balanced scorecard–measures that drive performance, Harv. Bus. Rev. 70 (1) (1992) 71–79.

[35] C.L. Yang, R.H. Huang, M.T. Ho, Multi-criteria evaluation model for a software development project, in: Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, Hong Kong, China, 2009.

[36] T.L. Saaty, The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw-Hill, 1980.

[37] C.A. Bana e Costa, J.C. Vansnick, A critical analysis of the eigenvalue method used to derive priorities in AHP, Eur. J. Oper. Res. 187 (3) (2008) 1422–1428, https://doi.org/10.1016/j.ejor.2006.09.022.

[38] J.S. Dyer, Remarks on the analytic hierarchy process, Manag. Sci. 36 (3) (1990) 249–258, https://doi.org/10.1287/mnsc.36.3.249.

[39] P. Goodwin, G. Wright, Decision Analysis for Management Judgment, 5th ed., John Wiley & Sons, 2014.

[40] V. Belton, T.J. Stewart, Multiple Criteria Decision Analysis: An Integrated Approach, Kluwer Academic Publishers, 2002.

[41] R.L. Keeney, D. von Winterfeldt, Practical value models, in: W. Edwards, R. F. Miles Jr., D. von Winterfeldt (Eds.), Advances in Decision Analysis: From Foundations to Applications, Cambridge University Press, 2007, pp. 232–252.

[42] J.S. Dyer, J.E. Smith, Innovations in the science and practice of decision analysis: the role of management science, Manag. Sci. 67 (9) (2020) 5364–5378, https://doi.org/10.1287/mnsc.2020.3652.

[43] J.E. Smith, J.S. Dyer, On (measurable) multiattribute value functions: an expository argument, Decis. Anal. 18 (4) (2021) 247–256, https://doi.org/10.1287/deca.2021.0435.

[44] J.S. Dyer, R.K. Sarin, Measurable multiattribute value functions, Oper. Res. 27 (4) (1979) 810–822, https://doi.org/10.1287/opre.27.4.810.

[45] R.L Keeney, Developing objectives and attributes, in: W. Edwards, R.F. Miles Jr., D. von Winterfeldt (Eds.), Advances in Decision Analysis: From Foundations to Applications, Cambridge University Press, 2007, pp. 104–128.

[46] B. Fasolo, C.A. Bana e Costa, Tailoring value elicitation to decision makers' numeracy and fluency: expressing value judgments in numbers or words, Omega 44 (0) (2014) 83–90, https://doi.org/10.1016/j.omega.2013.09.006.

[47] C.A. Bana e Costa, E.C. Corrêa, J.M. De Corte, J.C. Vansnick, Facilitating bid evaluation in public call for tenders: a socio-technical approach, Omega 30 (3) (2002) 227–242, https://doi.org/10.1016/S0305-0483(02)00029-4.

[48] R.L. Keeney, H. Raiffa, Decisions With Multiple Objectives: Preferences and Value Tradeoffs, John Wiley & Sons, 1976.

[49] W. Edwards, F.H. Barron, SMARTS and SMARTER: improved simple methods for multiattribute utility measurement, Organ. Behav. Hum. Decis. Process. 60 (3) (1994) 306–325, https://doi.org/10.1006/obhd.1994.1087.

[50] C.A. Bana e Costa, J.C. Vansnick, MACBETH – An interactive path towards the construction of cardinal value functions, Int. Trans. Oper. Res. 1 (4) (1994) 489–500, https://doi.org/10.1016/0969-6016(94)90010-8.

[51] C.A. Bana e Costa, J.M. De Corte, J.C. Vansnick, MACBETH, Int. J. Inf. Technol. Decis. Mak. 11 (2) (2012) 359–387, https://doi.org/10.1142/S0219622012400068.

[52] C.A. Bana e Costa, J.M. De Corte, J.C. Vansnick, On the mathematical foundations of MACBETH, in: S. Greco, M. Ehrgott, J.R. Figueira (Eds.), Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, 2016, pp. 421–463, https://doi.org/10.1007/978-1-4939-3094-4_11.

[53] W. Edwards, How to use multiattribute utility measurement for social decisionmaking, IEEE Trans. Syst. Man Cybern. 7 (5) (1977) 326–340, https://doi.org/10.1109/TSMC.1977.4309720.

[54] D. von Winterfeldt, W. Edwards, Decision Analysis and Behavioral Research, Cambridge University Press, 1986.

[55] C.W. Kirkwood, Strategic Decision Making: Multiobjective Decision Analysis with Spreadsheets, Duxbury Press, 1997.

[56] L.I. Assalaarachchi, M.P.P. Liyanage, C. Hewagamage, A framework of critical success factors of cloud-based project management software adoption, Int. J. Inf. Syst. Proj. Manag. 13 (2) (2025) e4, https://doi.org/10.12821/ijispm130204.

[57] N. Pinheiro, J. Vrajão, I. Moura, Success factors of public sector information systems projects in developing countries, Sustain. Futures 10 (2025) (2025) 101095, https://doi.org/10.1016/j.sftr.2025.101095.

[58] J. Jayakody, W. Wijayanayake, Critical success factors for DevOps adoption in information systems development, Int. J. Inf. Syst. Proj. Manag. 11 (3) (2023) 60–82, https://doi.org/10.12821/ijispm110304.

[59] K. Schwaber, J. Sutherland, The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game, scrumguides.org, 2020. https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf.

[60] M. Jovanovic, A.L. Mesquida, A. Mas, R. Colomo-Palacios, Agile transition and adoption frameworks, issues and factors: a systematic mapping, IEEE Access 8 (2020) (2020) 15711–15735, https://doi.org/10.1109/ACCESS.2020.2967839.

[61] V. Henriquez, J.A. Calvo-Manzano, A.M. Moreno, T. San Feliu, Agile governance practices by aligning CMMI V2.0 with portfolio SAFe 5.0, Comput. Stand. Interfaces 91 (2025) (2025) 103881, https://doi.org/10.1016/j.csi.2024.103881.

[62] V. Ferretti, G. Montibeller, Key challenges and meta-choices in designing and applying multi-criteria spatial decision support systems, Decis. Support Syst. 84 (2016) 41–52, https://doi.org/10.1016/j.dss.2016.01.005.

[63] L.D Phillips, Decision conferencing, in: W. Edwards, R.F. Miles Jr., D. von Winterfeldt (Eds.), Advances in Decision Analysis: From Foundations to Applications, Cambridge University Press, 2007, pp. 375–399.

[64] T.Y. Chen, H.F. Chang, Critical success factors and architecture of innovation services models in data industry, Expert Syst. Appl. 213 (2023) 119014, https://doi.org/10.1016/j.eswa.2022.119014.

[65] C.M. Smith, D. Shaw, The characteristics of problem structuring methods: a literature review, Eur. J. Oper. Res. 274 (2) (2019) 403–416, https://doi.org/10.1016/j.ejor.2018.05.003.

[66] M. Marttunen, J. Lienert, V. Belton, Structuring problems for multi-criteria decision analysis in practice: a literature review of method combinations, Eur. J. Oper. Res. 263 (1) (2017) 1–17, https://doi.org/10.1016/j.ejor.2017.04.041.

[67] C.A. Bana e Costa, J.C. Lourenço, M.P. Chagas, J.C. Bana e Costa, Development of reusable bid evaluation models for the Portuguese Electric Transmission Company, Decis. Anal. 5 (1) (2008) 22–42, https://doi.org/10.1287/deca.1080.0104.

[68] D. Clegg, R. Barker, Case Method Fast-Track: A RAD Approach, Addison-Wesley Longman Publishing, 1994.

[69] M. Weber, Decision making with incomplete information, Eur. J. Oper. Res. 28 (1) (1987) 44–57, https://doi.org/10.1016/0377-2217(87)90168-8.

[70] C.A. Bana e Costa, P. Vincke, Measuring credibility of compensatory preference statements when trade-offs are interval determined, Theory Decis. 39 (2) (1995) 127–155, https://doi.org/10.1007/BF01078981.

[71] L.D. Phillips, A theory of requisite decision models, Acta Psychol. 56 (1–3) (1984) 29–48, https://doi.org/10.1016/0001-6918(84)90005-2.

[72] J. Pereira, J. Varajão, N. Takagi, Evaluation of information systems project success–insights from practitioners, Inf. Syst. Manag. (2021) 1–18, https://doi.org/10.1080/10580530.2021.1887982.

[73] N. Takagi, J. Varajão, ISO 21502 and Success Management: A Required Marriage in Project Management, SAGE Open, 2025, pp. 1–11, https://doi.org/10.1177/21582440251355046. *July-September*.

[74] F.A.F. Ferreira, S.P. Santos, Two decades on the MACBETH approach: a bibliometric analysis, Ann. Oper. Res. 296 (1) (2021) 901–925, https://doi.org/10.1007/s10479-018-3083-9v.

[75] J. Varajão, L. Lopes, A. Tenera, Framework of standards, guides and methodologies for project, program, portfolio, and PMO management, Comput. Stand. Interfaces 92 (2025) (2025) 103888, https://doi.org/10.1016/j.csi.2024.103888.

[76] I.I. Mitroff, T.R. Featheringham, On systemic problem solving and the error of the third kind, Behav. Sci. 19 (6) (1974) 383–393, https://doi.org/10.1002/bs.3830190605.

[77] J. Varajão, Success Management as a PM knowledge area – work-in-progress, Procedia Comput. Sci. 100 (2016) (2016) 1095–1102, https://doi.org/10.1016/j.procs.2016.09.256.

[78] Y. Kong, N. Zhang, Z. Duan, B. Yu, Collaboration with generative AI to improve requirements change, Comput. Stand. Interfaces 94 (2025) (2025) 104013, https://doi.org/10.1016/j.csi.2025.104013.