Multi-criteria decision-making approach for modular ERP sorting problems

Silvia Carpitella[1], Antonella Certa[2], Joaquín Izquierdo[3], Marco La Cascia[2]

[1] Department of Decision-Making Theory - Institute of Information Theory and Automation, Czech Academy of Sciences, Prague, Czech Republic

[2] Department of Engineering, University of Palermo, Palermo, Italy

[3] Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Valencia, Spain

Abstract. Implementing Enterprise Resource Planning (ERP) systems is currently recognised as best practice with wide associated possibilities of business improvement for companies. Integrating these kinds of systems with business processes in the most efficient way requires to endeavour as much as possible simplifications for final users, which can be pursued by optimising crucial software characteristics. The present paper proposes a novel multi-criteria decision-making (MCDM) approach to deal with such an issue. Specifically, the ELECTRE (ELimination Et Choix Traduisant la REalité) TRI technique is suggested to assign ERP modules into predefined and ordered categories according to maintainability and usability, which are useful drivers in evaluating which module of an ERP software should be enhanced with priority. The results prove to have whole beneficial impact on system performance with relation to a case study: improvement evaluations will be identified based on the classes the ERP modules are assigned to by the method.

Keywords: Enterprise Resource Planning; maintainability; usability; MCDM; ELECTRE TRI

1. Introduction and literature review

1.1. Enterprise Resource Planning Systems

Optimising the process of enterprise resource planning is a cutting-edge and lively business challenge. Optimisation should be calibrated based on the degree of organisational complexity (Rodríguez et al., 2020) proper of the specific entity of reference, and according to established programs of business model innovation. Companies are currently moving towards the implementation of innovative IT business solutions capable of effectively managing their available resources. To this end, companies develop projects and processes lifecycles (Aboabdo et al., 2019), along with the relevant information flow via the Internet (Chen et al., 2015), as a part of their global business strategies (Gupta and Kohli, 2006). In this context, Enterprise Resource Planning (ERP) is an enterprise type of software incorporating relevant business functions and entities into a centralised comprehensive system towards outcomes optimisation (Chofreh et al., 2020). The critical role played by ERP as a key

strategic tool improving and supporting core business activities is recognised by the existing literature (Nofal and Yusof, 2013; Baykasoğlu and Gölcük, 2017; Lee and Wang, 2019) and, as asserted by Costa et al. (2016), the number of ERP users has tremendously increased, generating investments of millions of dollars over the last decades (Ruivo et al., 2020).
The main business areas typically integrated by ERP systems are detailed in Figure 1.

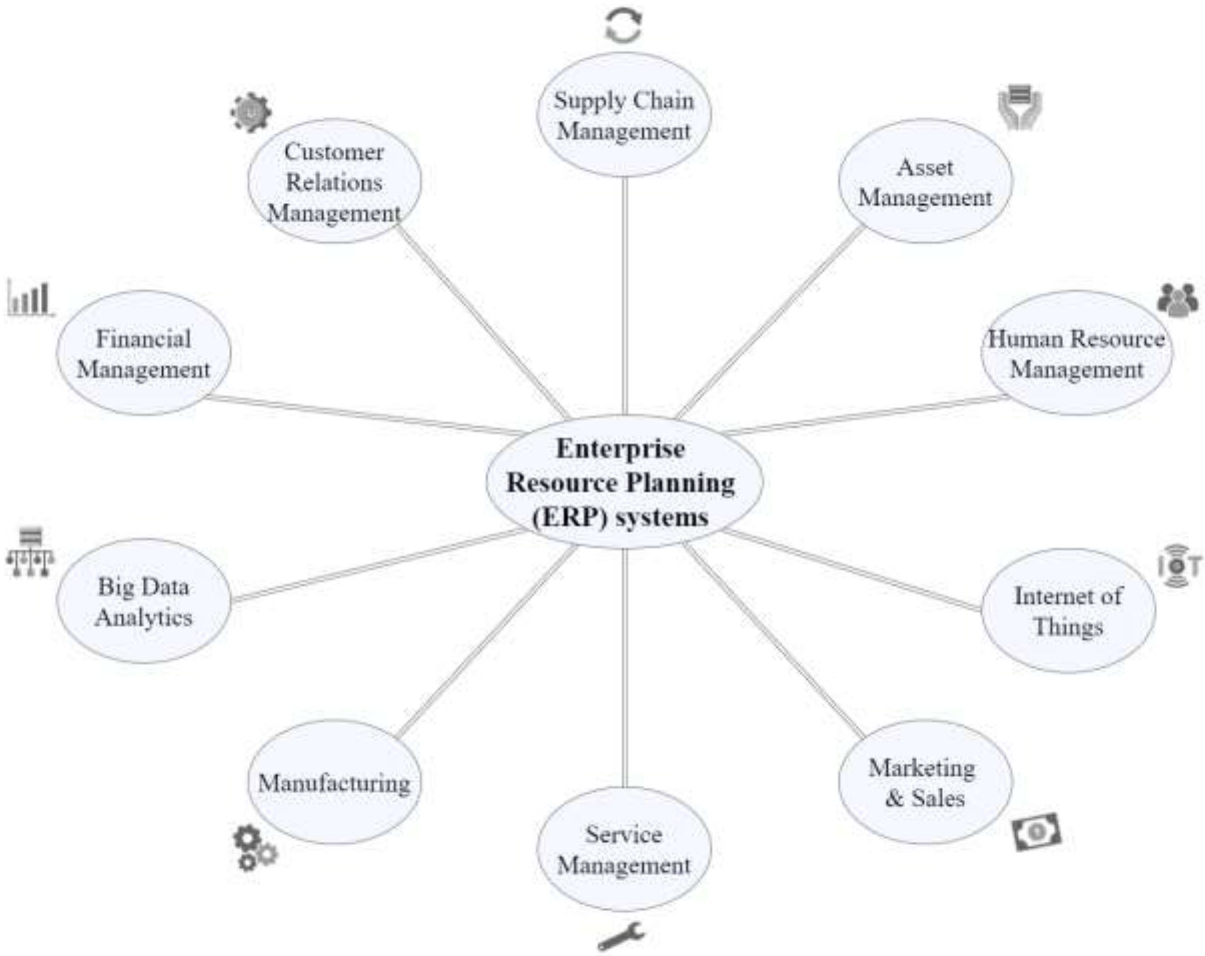

Figure 1. Main strategic areas integrated by ERP systems

This is the main reason why relevant risks related to ERP implementation and management should be taken into account to fully exploit systems' performance. An empirical survey led by Chang et al. (2015) based on data collected from ERP experts recognises the lack of support management and assistance as a main risk for successful ERP implementation. In order to improve such an aspect, a global analysis related to system modules should be accomplished by highlighting priorities of improvement above all based on maintainability (Gupta and Chug, 2020) and usability (Garousi et al., 2020) properties.

To the best of the authors' knowledge, the problem of sorting ERP modular systems is not still exhaustively covered by the existing literature, remaining an open issue. Maintainability and usability aspects are usually not effectively integrated in maintenance approaches carried out by software companies, so that sorting ERP system modules into predefined categories according to maintainability and usability criteria may offer perspectives of improvement for the final users and the scientific community.

1.2. Multi-criteria decision-making approaches and evaluation criteria

As reported by Chen et al. (2015), various MCDM methods (Greco et al., 2016; Ishizaka and Nemery, 2013) as well as mathematical programming approaches have been already proposed in the literature for evaluating and comparing ERP systems. Such authors as Karsak and Özogul (2009), Xu (2012) and Al-Rawashdeh et al. (2014) have, for instance, applied the Analytic Hierarchy Process (AHP) (Saaty, 1980) for ERP system selection and implementation. Similar applications in the field have been carried out by such methods as Data Envelopment Analysis (DEA) (Parthasarathy and Sharma, 2016), Artificial Neural Network (ANN) (Yazgan et al., 2009), Analytic Network Process (ANP) (Hallikainen et al., 2009; Lin et al., 2011), and other hybrid MCDM procedures (Bernroider and Stix, 2013; Gürbüz at al., 2012; Kilic et al., 2015). However, to the best of the authors' knowledge, the assignment of ERP modules to ordered categories according to such criteria as maintainability and usability has not been undertaken so far. In this regard, among the wide variety of available sorting models, Barak and Mokfi (2019) support the effectiveness of a MCDM-based evaluation.

ELECTRE (ELimination Et Choix Traduisant la REalité) TRI (Roy, 1991; Roy and Bouyssou, 1993) is one of the best known methods for sorting into predefined categories using preference relations, indicated as the most popular in the classification field by Doumpos et al. (2009). It enables the allocation of a set of alternatives into predetermined categories (Ramezanian, 2019), confirming to be a suitable technique for our analysis. Several versions of ELECTRE TRI have been proposed in the literature. Apart from the ELECTRE TRI-B herein proposed and initially developed by Yu (1992), other recent methods are ELECTRE TRI-C (Almeida-Dias et al., 2010), ELECTRE TRI-nC (Almeida-Dias et al., 2012) and ELECTRE TRI-nB (Fernández et al., 2017). From now on we are going to generically refer to the methodology we are proposing as ELECTRE TRI instead of ELECTRE TRI-B for the sake of brevity. ELECTRE TRI has been widely applied in the literature to support various application fields, such as information technology (Siskos et al., 2007), water distribution networks (Malekmohammadi et al., 2011; Brentan et al., 2020), project risk management (Certa et al., 2016), among others. Let us just cite a few examples. Liu and Ming (2019) propose the method to support the risk evaluation process related to complex smart product-service systems

(PPS). Sánchez-Lozano et al. (2016) apply the technique to support a Geographical Information System (GIS) in selecting the best location to deploy solar photovoltaic farms. Corrente et al. (2016) apply the hierarchical version of the ELECTRE TRI method to classify 223 projects regarding roads, ports and airports in the Balkans area.

In any case, selecting representative criteria to carry on the ELECTRE TRI application is fundamental to get successful outcomes. With relation to the problem analysed by the present paper, maintainability and usability are among the fundamental quality attributes of software engineering (Alsolai and Roper, 2020). This is why we proceed to consider them as main drivers for modular ERP system assignment. Maintainability and usability, respectively, refer to the degree to which an application can be understood, repaired, or enhanced during the software maintenance process (Gupta and Chug, 2020), and to the degree to which a software can be effectively used by specified users in a quantified context of use (Lee et al. 2019). These are pivotal aspects to be taken into account for effectively accomplishing activities of software maintenance, which is considered as a critical process over the life-cycle of industrial system applications (López and Salmeron, 2012). When leading such a kind of activities, some modifications (aimed at globally enhancing the mentioned criteria and making them more adherent to a changing environment) may be implemented on software, including corrections, bug fixing, performance improvements, updates of functional requirements and specifications (Gupta et al., 2013). To such an aim, the proposed approach easily enables to identify whether a software module needs to be modified with priority.

1.3. Research objectives and paper structure

The research objectives of the present paper are the following:

a) to adopt maintainability and usability as main drivers to perform modular ERP systems evaluation aimed at improving software performance and results for final users;
b) to propose, as a novel idea, the MCDM method ELECTRE TRI to assign ERP modules into predefined and ordered classes according to maintainability and usability criteria.

We claim this approach enables to effectively support the analyst in identifying which modules have a stronger and more critical impact than others on system performance on the basis of the classes to which those modules are assigned by the mentioned MCDM method.

The paper is structured as follows. Section 2 provides relevant aspects about software maintainability and usability along with the reasons why we chose them as the main analysis drivers, and the description of the ELECTRE TRI technique. A case study on ERP system modules classification is discussed in section 3, and practical implications for companies are detailed. Lastly, section 4 provides conclusions and possible future developments of the present research.

2. Materials and Methods

2.1. Maintainability and usability standards

The international standard ISO/IEC 9126 "Software Engineering - Product Quality", now replaced by ISO/IEC 25010 (2011), which has been reviewed and confirmed in the year 2017, provides a comprehensive set of standards and guidelines for the implementation of a software quality model.

The first part of the mentioned standard establishes a quality model as classified according to six main categories, namely functionality, reliability, usability, efficiency, maintainability, and portability. The first attribute is considered as a functional requirement, whereas the other five attributes are quality requirements, playing a fundamental part in software engineering.

The formal justification of the choice of usability and maintainability as main drivers with respect to the other characteristics to deal with the ERP module sorting problem is provided next.

Usability is considered as a key quality attribute of software systems from the industrial perspective (Raza et al., 2011), and diverse works of research underline the usage of a system supporting user activities among the most important objectives of software engineering (Winter et al., 2007). Measuring usability is strategic because it allows to quantify how effectively users can interact with software systems. As asserted by Bødker and Sundblad (2007), users' needs and expectations should be fully understood by developers and addressed with full motivation, in order to maximise the popularity and adoption of software products, and to minimise the risk of poor user friendliness. Indeed, any product that is able to flawlessly performs its primary technical function without guaranteeing successful interactions with users has failed. This last consideration particularly highlights the primary role of usability with respect to the other characteristics of functionality, reliability and efficiency defined by the standard. This is the reason why we consider important the integration of usability as a main criterion for sorting ERP software modules.

With respect to the second main driver, maintainability, our choice is justified by the evidence that it is significantly related to all the remaining characteristics. Software maintainability analyses should be led at the levels of component, subsystem and whole system to directly and simultaneously enhance its functionality, reliability and efficiency. Maintainability is considered as a crucial prerequisite for lifelong evolution of systems on the basis of efficient updating of their related software (Vogel-Heuser and Ocker, 2018), and is derived from particular source code metrics as indicators (Hegedűs et al., 2018). As observed by Saraiva et al. (2015), a considerable number of product metrics has been proposed for software maintainability assessment in the literature, what poses a decision-making challenge for practitioners and researchers with relation to gain a consolidated overview about available and practically beneficial software metrics (Bouwers et al., 2014). Atalag et al. (2014) recognise as even tiny improvements in maintainability can be translated into considerable cost savings and product reliability increase. The same authors affirm that software continues to evolve throughout its life-cycle, due to the realisation of both maintenance and planned fashion processes aimed at progressively releasing versions of a more effective product.

The reasons why usability and maintainability have been chosen as specific drivers to deal with the ERP categorisation problem are then clear. We further specify that the other software characteristics are not contemplated in the present research as drivers of analysis for two main reasons: 1) such aspects as functionality, reliability and efficiency are of secondary importance with respect to usability (Kortum and Bangor, 2013); 2) functionality, reliability, efficiency and portability are directly connected to maintainability, whose enhancement would then directly contribute to increase all the remaining aspects (Coleman et al., 1994).

Table 1 presents their definition along with the related sub aspects according to the standard of reference. Note that each sub driver described in Table 1 is related to one or more usability/maintainability metrics referring to the code-level properties, as shown by such authors as Sauro and Kindlund (2005) and Heitlager et al. (2007). These metrics, detailed in Figure 2, are denoted as “criteria” since they will constitute the set of evaluation criteria for the ELECTRE TRI application.

Table 1. Definition of usability and maintainability according to the international standard ISO/IEC 9126

Drivers	Sub drivers	Description
Usability “*A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.*”	Understandability	It is an indicator of how easily product contents can be understood, enabling the user to determine if the software is adequate for his/her own purposes or not.
	Learnability	It is a measure of how much the commitment from users in learning how to use the software can be reduced.
	Operability	It expresses the capability of enabling users to adapt the application to their own purposes and, simultaneously, to control usage procedures.
	Attractiveness	It estimates the software capability to be pleasant to use and its appeal degree for the final user.

Maintainability "*A set of attributes that bear on the effort needed to make specified modifications.*"	Analysability	It indicates how easily the software can be diagnosed for analysing potential deficits, as well as indicating parts needing to be updated in order to identify root causes of failures.
	Changeability	It measures how much the software enables the implementation of specific modification, giving an idea about the effort required to modify the system.
	Stability	It refers to the capability of minimising undesirable effects caused by software modifications, being an indicator of sensitivity to changes for a given system.
	Testability	It synthetises the capability of validating modifications on the software product, charactering the effort needed to test changes in the system.

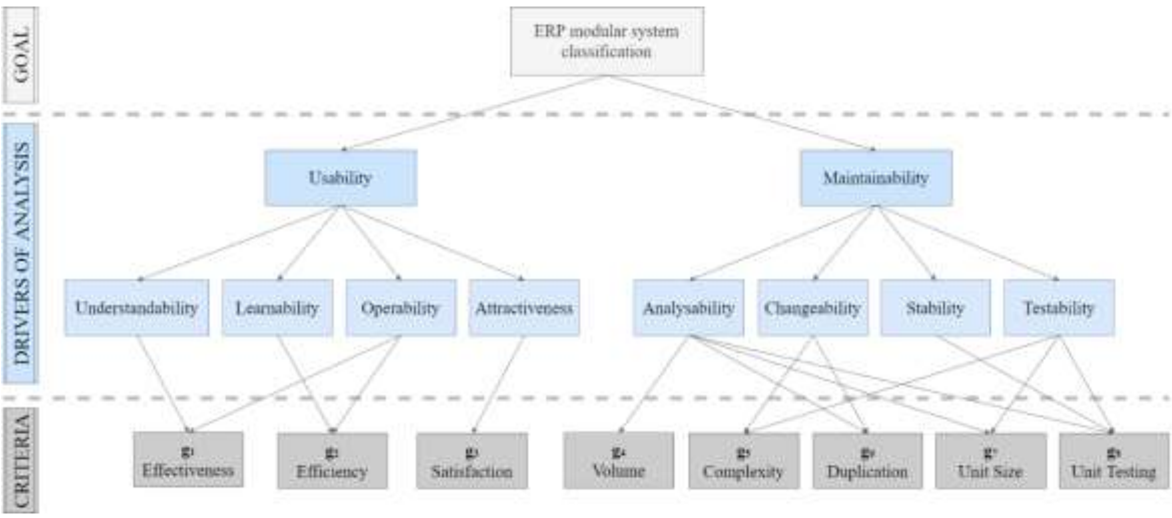

Figure 2. Elements of the problem and criteria associated to usability and maintainability

Usability metrics (Pedroli et al., 2018) allow to quantitatively express the main parameters globally defining software usability. Effectiveness (g_1) is defined as the degree of accuracy in achieving determined objectives; efficiency (g_2) refers to resources used in relation to the results achieved; and, lastly, satisfaction (g_3) indicates how much the product meets users' expectations in terms of their physical, cognitive and emotional responses. Sauro and Kindlund (2005) suggest that effectiveness, efficiency and satisfaction are respectively related to the following parameters: task completion, task time and satisfaction score, the last one measurable by treating results of proper surveys.

In its turn, maintainability metrics (Heitlager et al., 2007) permit to quantitatively evaluate the main characteristics generally describing software maintainability, strongly impacting on software quality (Baggen et al., 2012). Specifically, g_4 is measured as the overall number of lines of source code, g_5 represents the number of linearly independent paths through the source code, g_6 refers to the so-called code cloning as density of source code duplication, g_7 indicates the lines of code units and, lastly, g_8 is based on the coverage of the application by unit tests.

Using the defined metrics, the assignment procedure carried out by ELECTRE TRI will support in identifying the criticality degree related to each software module with relation to software quality. This will enable us to evaluate those modules, the ones belonging to the worst class, in need of being updated/maintained with priority to significantly improve system quality.

2.2. The ELECTRE TRI method for the ERP modules sorting problem

ELECTRE TRI is an outranking-based technique applied to sorting categorisation decision-making problems. This method does not require alternatives to be pairwise compared among them, since they are not in competition with each other but need to be assigned to classes based on their capability to match specific conditions. This is the reason why assignments to classes are performed by pairwise comparing alternatives with certain profiles of reference, which will have to be carefully defined.

The ELECTRE TRI belongs to the ELECTRE family of MCDM methods. They are also known as outranking methods since a specific outranking relation has to be developed as a basis for their application (Lourenço and Costa, 2004; Figueira et al., 2010). An outranking relation expresses particular conditions existing between pairs of alternatives or, in the case of the ELECTRE TRI method, between alternatives and reference profiles. Such a kind of relation is based on concordance/discordance principles, which consist in verifying the concordance among criteria about the fact that a given alternative outranks another alternative (or reference profile) along with the discordance among criteria that this assertion can be rejected. A generic relation can underline conditions of indifference, preference or incomparability. In the first case, an alternative outranks a reference profile and vice versa; in the second case, an alternative outranks a reference profile and not vice versa; in the last case, alternative and reference profile diverge too much, so that they cannot be compared. These conditions are expressed by fixing proper numerical thresholds (Carpitella at al., 2018). Specifically, an indifference threshold is the maximal difference justifying an indifference between two alternatives; a preference threshold is the minimal performance difference validating a strong preference of one alternative over another, whereas a veto threshold indicates the minimal performance difference invalidating the truth of an outranking relation.

Establishing threshold values (Roy et al., 2014) represents an extremely important issue for ELECTRE TRI, having direct impact on the classification results (Dias and Mousseau, 2006). As explained by Mousseau et al. (2000), threshold values have to be established by the analyst in order to calibrate the method in adherence with the particular problem under study. Larger values can be first simulated for thresholds by leading various attempts and then progressively reduced until considered appropriate for each criterion. Among the main advantages of ELECTRE TRI, there is the possibility to consider vagueness affecting experts' judgments involving the assignment procedure, by associating an outranking relation with a quantity between 0 and 1 measuring the so-called degree of credibility. The following two main stages have to be developed to carry out the ELECTRE TRI methodology (Figueira et al., 2013, 2016). The first stage consists in defining an outranking relation between pairs of alternatives and reference profiles through the calculation of concordance and discordance indices, carried out by following the same computation procedure of ELECTRE III (Roy, 1991), another method of the ELECTRE family (Govindan and Jepsen, 2016). The second stage consists in properly assigning alternatives to classes (on the basis of the outranking relation established during the previous stage) by means of two possible ways, namely the pessimistic and optimistic procedures. Figure 3 illustrates the whole framework of the method.

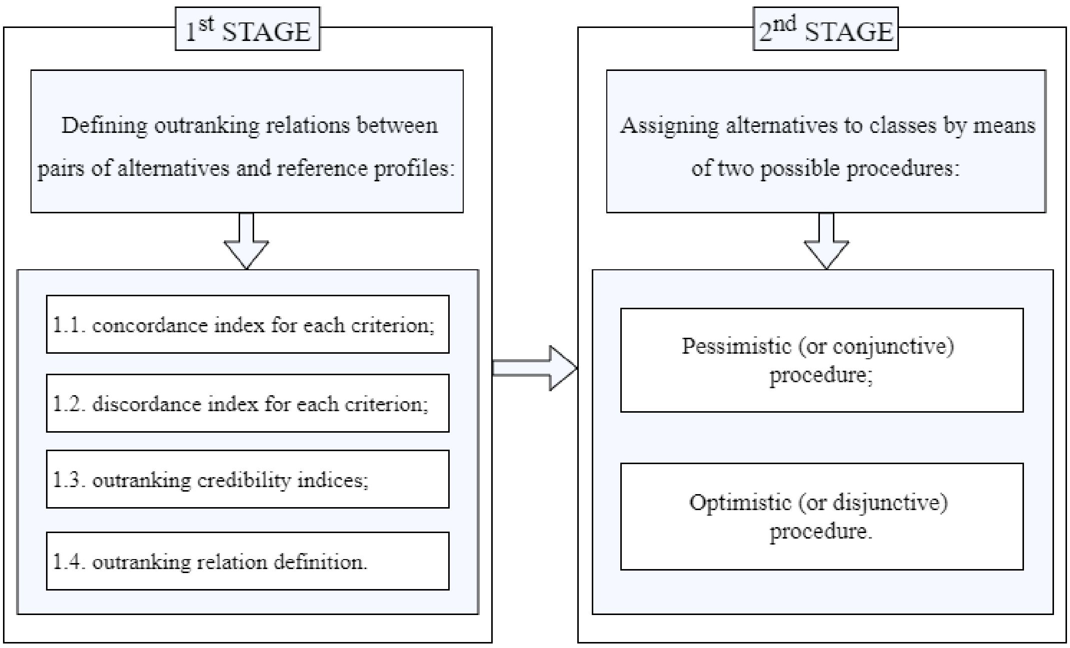

Figure 3. Framework of the ELECTRE TRI procedure

ELECTRE TRI requires the previous definition of ordered classes without any intersection among the related reference profiles. Each reference profile simultaneously represents the upper reference profile for a class and the lower reference profile for the next class. Reference profiles can be directly provided by the analyst or a decision-making team (Cailloux et al., 2012), or also by means of specific elicitation techniques admitting indirect preference information (Mousseau and Słowiński, 1998; Mousseau and Ngo The, 2002). Figure 4 exemplifies the definition of five ordered classes delimited by four reference profiles with relation to five generic criteria.

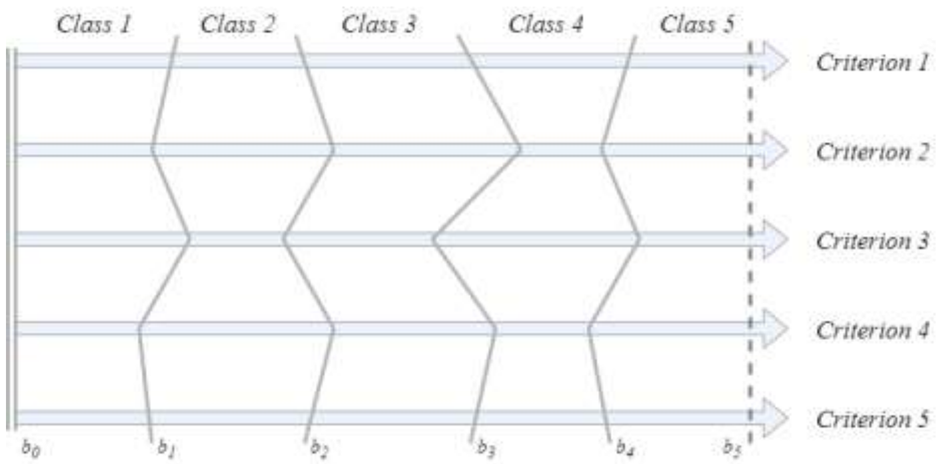

Figure 4. Classes and reference profiles representation

With those preliminary considerations about the ELECTRE TRI technique in mind, the following input data are necessary to proceed with the application:

- set of criteria g_j, (j = 1, …, J), relevant to the decision-making problem under analysis; and criteria weights w_j, expressing their relative importance;
- set of reference profiles b_k, (k = 1, …, K), corresponding to specific evaluations for each criterion j, and delimited by values $b_0^{(j)} < \ldots < b_{K+1}^{(j)}$ (noted without super indexes in Figure 4);
- set of classes C_h, (h = 1, …, K+1), determined by the K reference profiles;
- set of alternatives A_i, (i = 1, …, I), along with their related evaluations $g_j(A_i)$ under each criterion;
- cutting value $\lambda \in]0.5, 1]$, a threshold value needed to complete the first stage of the ELECTRE TRI procedure;
- indifference, strong preference and veto thresholds characterising relations between sets of pairs and respectively indicated by the notations q_j, p_j, and v_j.

Once collected all the necessary input data, we are going to detail the two explained stages as follows.

1st STAGE: establishing an outranking relation **S** comparing each alternative with limits of classes, i.e. with the reference profiles. This stage is made of four intermediate steps.

1.1. Calculation of concordance indices for each criterion. Each alternative A_i has to be pairwise compared with all the defined reference profiles b_k and concordance indices, indicated as $C_j(A_i, b_k)$ have to be calculated for each criterion g_j by means of the following formula:

$$C_j(A_i, b_k) = \begin{cases} 1 & \text{if } g_j(b_k) - g_j(A_i) \leq q_j \\ \frac{g_j(A_i) - g_j(b_k) + p_j}{p_j - q_j} & \text{if } q_j < g_j(b_k) - g_j(A_i) < p_j. \\ 0 & \text{if } g_j(b_k) - g_j(A_i) \geq p_j \end{cases} \quad (1)$$

From the previous calculation of concordance indices related to each criterion, it will be possible to derive the aggregated concordance index $C(A_i, b_k)$ by aggregating and weighting the indices as follows:

$$C(A_i, b_k) = \frac{\sum_{j=1}^{J} w_j \cdot C_j(A_i, b_k)}{\sum_{j=1}^{J} w_j}. \quad (2)$$

1.2. Calculation of discordance indices for each criterion by using the following formula:

$$D_j(A_i, b_k) = \begin{cases} 1 & \text{if } g_j(b_k) - g_j(A_i) > v_j \\ \frac{g_j(b_k) - g_j(A_i) - p_j}{v_j - p_j} & \text{if } p_j < g_j(b_k) - g_j(A_i) \leq v_j. \\ 0 & \text{if } g_j(b_k) - g_j(A_i) \leq p_j \end{cases} \quad (3)$$

1.3. Calculation of outranking credibility indices through the following equation:

$$\sigma(A_i, b_k) = \prod_{j \in F} \frac{1 - D_j(A_i, b_k)}{1 - C(A_i, b_k)}, \quad (4)$$

where $F = \{j: D_j(A_i, b_k) > C(A_i, b_k)\}$; $\sigma(A_i, b_k) = C(A_i, b_k)$ otherwise. If the veto threshold is not defined for any criterion, then, the credibility index $\sigma(A_i, b_k)$ is equal to the aggregated concordance index, $C(A_i, b_k)$. After having been computed, a fuzzy outranking relation based on credibility indices will have to be translated into a crisp relation.

1.4. Definition of the specific kind of outranking relation by using the cutting level λ, which represents the threshold value for $\sigma(A_i, b_k)$ to accept the hypothesis that A_i outranks b_k. Liu and Ming (2019) quote Merad et al. (2004) circa the value of λ, affirming that it is comprised in the interval [0.5, 1]. In particular, the values of $\sigma(A_i, b_k)$, $\sigma(b_k, A_i)$ and λ determine the preference relation between A_i and b_k (Mousseau et al., 2001):.

- $\sigma(A_i, b_k) \geq \lambda$ and $\sigma(b_k, A_i) \geq \lambda \Rightarrow A_i$ **S** b_k and b_k **S** $A_i \Rightarrow A_i$ **I** b_k;
- $\sigma(A_i, b_k) \geq \lambda$ and $\sigma(b_k, A_i) < \lambda \Rightarrow A_i$ **S** b_k and not b_k **S** $A_i \Rightarrow A_i$ **P** b_k;
- $\sigma(A_i, b_k) < \lambda$ and $\sigma(b_k, A_i) \geq \lambda \Rightarrow$ not A_i **S** b_k and b_k **S** $A_i \Rightarrow b_k$ **P** A_i;

- $\sigma(A_i, b_k) < \lambda$ and $\sigma(b_k, A_i) < \lambda \Rightarrow$ not A_i **S** b_k and not b_k **S** $A_i \Rightarrow A_i$ **R** b_k;

where **S** indicates the outranking relation (specifically, A_i **S** b_k indicates that alternative i is at least as good as reference profile k) and **I**, **P** and **R** respectively indicate indifference, strong preference and incomparability relation.

2nd STAGE: assigning alternatives to classes by means of the pessimistic and the optimistic rules.
Pessimistic (or conjunctive) procedure: alternative A_i is assigned to the class C_k for which the stopping condition that A_i **S** b_k, that is A_i is at least as good as profile b_k, is verified. The pessimistic assignment begins from the upper value limiting reference profiles defining classes and is carried out through two steps:

- sequentially comparing each alternative with the limits of classes. In other words, A_i is successively compared to profiles defining classes until verifying the condition A_i **S** b_k;
- assigning alternative A_i to class C_{k+1}.

Optimistic (or disjunctive) procedure: alternative A_i is assigned to the class C_k for which the stopping condition b_k **P** A_i is verified, that is reference profile k needs to be preferred over alternative i. The optimistic assignment begins from the lower value limiting reference profiles defining classes and is carried out through two steps:

- sequentially comparing each alternative with the limits of classes. In other words, A_i is successively compared to profiles defining classes until verifying the condition b_k **P** A_i;
- assigning alternative A_i to class C_k.

Results derived from the two procedures do not necessarily coincide. Specifically, they will assign an alternative to the same class when, for each criterion, the alternative evaluation falls between two profiles defining that class. Otherwise, if a divergence exists, it indicates the presence of incomparability conditions between the alternative and one or more reference profiles. In such a case, the pessimistic procedure will assign the alternative to a lower class with respect to the optimistic procedure. On the whole, the pessimistic procedure should be preferred to the optimistic one, tending to assign alternatives to classes defined by a lower profile, what actually guarantees the achievement of more conservative results.

3. Case study and discussion of results

The present case study refers to a subprogram of an ERP platform integrating tools, services and functions aimed at supporting the sales business process of an existing industry operating in the South of Italy, whose core business consists in producing and commercialising various types of wines. Specifically, we analyse and classify five software modules by means of ELECTRE TRI according to the set of criteria g_j ($j = 1 \dots 8$) that are the metrics referring to the main drivers of usability and maintainability previously discussed in subsection 2.1. The mentioned five modules, denoted by A_i ($i = 1 \dots 5$), constitute the set of alternatives of the decision-making problem, and respectively refer to the management of the following activities: 1) order placement, 2) order scheduling, 3) order shipping, 4) order tracking, and 5) invoicing. Module A_1 (placement), corresponding to the task of order placement, includes all the aspects related to the entrance of customer data by means of the order management system. Module A_2 (scheduling), corresponding to the task of order scheduling, includes the inventory management as well as the aspects related to picking, packing and labelling. Modules A_3 (shipping) and A_4 (tracking), respectively corresponding to the tasks of order shipping and order tracking, are important for managing the shipping and the delivery process, by means of a dedicated system of carrier control. Lastly, module A_5 (invoicing), corresponding to the homonym task, refers to all the aspects related to payments processing and invoice emission.

The analysed modules are not totally independent from each other, since they share some type of input data and some of the related processes are mutually correlated. For example, it is clear that, when the placement fails, such a failure has a negative impact on the order scheduling which, in its turn, may also negatively impact the remaining processes. However, modules can be improved in a separate way, guaranteeing the functioning of other modules during restoring processes and posteriorly carrying out proper data synchronization. Table 2 provides the numerical evaluation of the alternatives under the abovementioned criteria, along with a set of related criteria weights.

Table 2. Alternatives vs criteria evaluation, and criteria weights

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
Weights → **Alternatives**	16.67%	16.67%	16.67%	10.00%	10.00%	10.00%	10.00%	10.00%
A_1: Placement	0.68	0.55	6.20	151887	13138	0.10	70	0.70
A_2: Scheduling	0.76	0.69	7.30	134667	10003	0.09	80	0.80
A_3: Shipping	0.83	0.78	7.70	99864	9798	0.08	60	0.60
A_4: Tracking	0.42	0.34	3.30	23557	16789	0.12	70	0.30
A_5: Invoicing	0.48	0.43	5.50	173671	14034	0.11	65	0.65

Input data of Table 2 has been collected with the help of the IT department of the company along with the responsible of the sales process, who is also the master user of the sales software under analysis. The meaning of this set of data is explained next.

The first three metrics representing usability, namely effectiveness (g_1), efficiency (g_2) and satisfaction (g_3), are respectively expressed as: task completion rate (the number of tasks successfully completed divided by the total number of tasks undertaken), time-based efficiency (the ratio of the time, in seconds, taken by the users who successfully completed the task, to the total time taken by all users), and task level satisfaction (measured on the basis of the NASA-TLX questionnaire: https://humansystems.arc.nasa.gov/groups/TLX/downloads/TLXScale.pdf).

The five metrics, representing maintainability, have been evaluated as follows: volume (g_4) as the number of lines of source code, complexity (g_5) as the number of linearly independent paths through the source code, duplication (g_6) as the density of source code duplication, unit size (g_7) as the percentage of lines of code units and, lastly, unit test (g_8) as the coverage of testing for the application. At this stage we assume usability and maintainability as having the same relative importance, what corresponds to a global weight of 50% for the three metrics representing usability, and a global weight of 50% for the five metrics representing maintainability. When it comes to preference directions, the three criteria representing usability and the last criterion referring to maintainability (namely g_1, g_2, g_3, g_8) have increasing preference directions, whereas the first four criteria related to maintainability (g_4, g_5, g_6, g_7) have decreasing preference directions.

The alternatives are then classified into four disjoint ordered classes identified (separated) by three reference profiles, b_k ($k = 1 \ldots 3$), given in the last three columns of Table 3. The first three columns show the indifference, preference and veto thresholds for each criterion. The three profiles simultaneously constitute the upper limits for classes C_1, C_2, C_3, and the lower limits for classes C_2, C_3, C_4. The classes have been ordered from C_1 to C_4 to express the transition from a condition of low performance to a condition of high performance for the ERP platform. The different classes highlight the level of quality of modules according to specific intervals of values assumed by the chosen criteria (herein referred to both maintainability and usability drivers). Classes are ordered in the following way: C_1, low performance; C_2, medium-low performance; C_3, medium-high performance; and C_4, high performance. Discrimination thresholds have been established by first setting larger values and progressively reducing them until considered as appropriate for each criterion. The veto threshold has been assumed as equal to the width of classes, whereas the preference and indifference thresholds respectively as a half and a quarter of the veto threshold, respectively. We specify that the difference between the evaluation of consecutive profiles on each criterion is always the same: $g_j(b_3) - g_j(b_2) = g_j(b_2) - g_j(b_1)$. In Table 3 we can observe as these differences have been set equal to the veto thresholds for each criterion.

Table 4 lastly presents the assignment of each software module to the defined classes according to the pessimistic and the optimistic procedures, achieved by varying the value of the cutting level λ from 0.5 to 1 and by considering as threshold values the ones reported in Table 3.

Table 3. Threshold definition and reference profiles

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
q_j	0.0625	0.0625	0.625	18750	1250	0.01	6.25	0.0625
p_j	0.125	0.125	1.25	37500	2500	0.02	12.5	0.125
v_j	0.25	0.25	2.50	75000	5000	0.04	25	0.25
b_1	0.25	0.25	2.50	225000	15000	0.12	75	0.25
b_2	0.50	0.50	5.00	150000	10000	0.08	50	0.50
b_3	0.75	0.75	7.50	75000	5000	0.04	25	0.75

Table 4. Assignment of modules to classes by means of the pessimistic and optimistic procedures

	PESSIMISTIC PROCEDURE				**OPTIMISTIC PROCEDURE**			
λ	C_1 **Low performance**	C_2 **Medium-low performance**	C_3 **Medium-high performance**	C_4 **High performance**	C_1 **Low performance**	C_2 **Medium-low performance**	C_3 **Medium-high performance**	C_4 **High performance**
0.50	A_4		A_1, A_2, A_3, A_5				A_4	A_1, A_2, A_3, A_5
0.55	A_4		A_1, A_2, A_3, A_5				A_4	A_1, A_2, A_3, A_5
0.60	A_4		A_1, A_2, A_3, A_5				A_4	A_1, A_2, A_3, A_5
0.65	A_4	A_1	A_2, A_3, A_5				A_4	A_1, A_2, A_3, A_5
0.70	A_4	A_1	A_2, A_3, A_5					A_1, A_2, A_3, A_4, A_5
0.75	A_4	A_1	A_2, A_3, A_5					A_1, A_2, A_3, A_4, A_5
0.80	A_4	A_1	A_2, A_3, A_5					A_1, A_2, A_3, A_4, A_5
0.85	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.90	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.95	A_4, A_3	A_1, A_5	A_2					A_1, A_2, A_3, A_4, A_5
1.00	A_4, A_3, A_5	A_1	A_2					A_1, A_2, A_3, A_4, A_5

Results reported in Table 4 have been eventually double checked and validated by means of the JElectre-v2.0 software for multi-criteria decision aid (https://sourceforge.net/projects/j-electre/files/). As it is possible to observe, results deriving from the pessimistic and the optimistic procedures diverge also for small values of the cutting level, and even more for λ > 0.65. Once overcome the value of 0.65, even alternative A_4, which is always assigned to the low performance class (C_1) by the pessimistic procedure, is assigned to the high performance class (C_4) by the optimistic rule. By comparing the two procedures, it is clearly preferable and more prudent relying on results obtained by means of the pessimistic rule. They indeed confirm to be more conservative, being alternatives assigned to classes defined by lower profiles with respect to the assignment carried out through the optimistic procedure. With special reference to the pessimistic procedure, results confirm to be quite robust by varying the cutting level. In particular, no variations are noticed in the interval [0.65, 0.80]. Out of this interval, in particular for λ < 0.65, alternative A_1 is upgraded from the medium-low performance class (C_2) and assigned to the medium-high performance class (C_3). Alternatives A_5 and A_3 leave the high performance class respectively for λ > 0.8 and λ > 0.95. However, one has to note that alternatives A_4 and A_2 are stable throughout the whole interval of variation of the cutting level, being respectively assigned to the low and to the medium-high performance classes (i.e. C_1 and C_3) with no variation whatsoever. These results give an immediate idea about which modules are in higher/lower need to be improved.

For the sake of completeness, aiming at analysing the influence of the discrimination thresholds, Tables 5 and 6 show the results obtained by setting different values in order to represent stricter (Table 5) and less strict (Table 6) scenarios with respect to the baseline test, whose results are given in Table 4. Values of veto threshold, respectively, equal 0.75 (Table 5) and 1.25 (Table 6) times the width of the analysed classes. It means that values of veto threshold respectively equal to 0.75 and 1.25 times the initial value assumed for the veto threshold on each criterion shown in Table 3 have been considered to lead the two tests. In both cases the values of preference and indifference thresholds (the last ones assumed as equal to a half and a quarter of the veto threshold) have been consequently adjusted.

Table 5. Assignment led by decreasing the veto (stricter scenario)

	PESSIMISTIC PROCEDURE				**OPTIMISTIC PROCEDURE**			
λ	**C_1 Low performance**	**C_2 Medium-low performance**	**C_3 Medium-high performance**	**C_4 High performance**	**C_1 Low performance**	**C_2 Medium-low performance**	**C_3 Medium-high performance**	**C_4 High performance**
0.50	A_4	A_1	A_2, A_3, A_5					A_1, A_2, A_3, A_4, A_5
0.55	A_4	A_1	A_2, A_3, A_5					A_1, A_2, A_3, A_4, A_5

0.60	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.65	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.70	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.75	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.80	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.85	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.90	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_4, A_5
0.95	A_4, A_3, A_5	A_1	A_2					A_1, A_2, A_3, A_4, A_5
1.00	A_1, A_3, A_4, A_5		A_2					A_1, A_2, A_3, A_4, A_5

Table 6. Assignment led by increasing the veto (less strict scenario)

	PESSIMISTIC PROCEDURE				**OPTIMISTIC PROCEDURE**			
λ	**C_1 Low performance**	**C_2 Medium-low performance**	**C_3 Medium-high performance**	**C_4 High performance**	**C_1 Low performance**	**C_2 Medium-low performance**	**C_3 Medium-high**	**C_4 High performance**
0.50	A_4		A_2, A_3, A_5	A_1			A_4	A_1, A_2, A_3, A_5
0.55	A_4		A_2, A_3, A_5	A_1			A_4	A_1, A_2, A_3, A_5
0.60	A_4		A_2, A_3, A_5	A_1			A_4	A_1, A_2, A_3, A_5
0.65	A_4		A_2, A_3, A_5	A_1			A_4	A_1, A_2, A_3, A_5
0.70	A_4		A_2, A_3, A_5	A_1			A_4	A_1, A_2, A_3, A_5
0.75	A_4		A_1, A_2, A_3, A_5				A_4	A_1, A_2, A_3, A_5
0.80	A_4		A_1, A_2, A_3, A_5				A_4	A_1, A_2, A_3, A_5
0.85	A_4	A_1	A_2, A_3, A_5					A_1, A_2, A_3, A_5
0.90	A_4	A_1, A_5	A_2, A_3					A_1, A_2, A_3, A_5
0.95	A_4, A_3	A_1, A_5	A_2					A_1, A_2, A_3, A_5
1.00	A_4, A_3	A_1, A_5	A_2					A_1, A_2, A_3, A_5

To complete the case study, we undertake a sensitivity analysis by varying the criteria weights, aimed at checking robustness of the results achieved by means of the pessimistic procedure reported in Table 4. The related results are synthetized in Table 7 and compared with the baseline scenario, in which equal weights had been considered for the two drivers, usability and maintainability. We specify that the same attempts may also be led by considering different values for the cutting level λ. In the present case, we report results obtained by fixing the value of λ to 0.8, that is the last value of the interval [0.65, 0.8] for which the pessimistic procedure of Table 4 does not give different outputs.

Specifically, the following five scenarios have been further considered:

- 1st scenario: 60% usability and 40% maintainability, with the following associated vector of criteria weights: $[0.20; 0.20; 0.20; 0.08; 0.08; 0.08; 0.08; 0.08]$;
- 2nd scenario: 70% usability and 30% maintainability, with the following associated vector of criteria weights: $[0.2333; 0.2333; 0.2333; 0.060; 0.060; 0.060; 0.060; 0.060]$;
- 3rd scenario: 40% usability and 60% maintainability, with the following associated vector of criteria weights: $[0.1333;\ 0.1333;\ 0.1333;\ 0.12;\ 0.12;\ 0.12;\ 0.12;\ 0.12]$;
- 4th scenario: 30% usability and 70% maintainability, with the following associated vector of criteria weights: $[0.10;\ 0.10;\ 0.10;\ 0.14;\ 0.14;\ 0.14;\ 0.14;\ 0.14]$.
- 5th scenario: 15% usability and 85% maintainability, with the following associated vector of criteria weights: $[0.05;\ 0.05;\ 0.05;\ 0.17;\ 0.17;\ 0.17;\ 0.17;\ 0.17]$.

By observing the results of Table 7, various practical considerations can be derived. First of all, the sensitivity analysis led on criteria weights confirms the robustness of the assignment procedure. In the case in which usability and maintainability have associated the same relative importance (baseline scenario) within the decision-making process, the module in need of being improved with priority is the fourth alternative (A_4), which is assigned to the low performance class, C_1, by the ELECTRE TRI technique.

Table 7. Assignment in each scenario considered by the sensitivity analysis

	Basis scenario	**1st scenario**	**2nd scenario**	**3rd scenario**	**4th scenario**	**5th scenario**
$\boldsymbol{A_1}$	C_2	C_2	C_2	C_2	C_2	C_3
$\boldsymbol{A_2}$	C_3	C_3	C_3	C_3	C_3	C_3
$\boldsymbol{A_3}$	C_3	C_3	C_3	C_3	C_3	C_3
$\boldsymbol{A_4}$	C_1	C_1	C_1	C_1	C_1	C_1
$\boldsymbol{A_5}$	C_3	C_3	C_3	C_3	C_2	C_2

This alternative corresponds to the module dedicated to the process of order tracking. Enhancing the module dedicated to the management of order tracking has direct benefits for all the involved stakeholders. The tracking module is indeed fundamental for all the phases of the order management process. First of all, confirming that orders have been correctly received and confirmed is useful to avoid potential losses of projected revenue. Moreover, knowing at any time the state of orders throughout the activities of preparing, shipping and delivery is fundamental to minimise operational delays and to organise how to receive orders in case of multiple delivery dates and/or destinations. An effective order tracking process should also provide proper reports to be shared among all the stakeholders so that keeping a trace of any relevant information related to orders can be possible. As a result, enhancing the main criticalities of the tracking module would entail a direct increase of the whole level of sales software performance. Such an improvement would directly lead to improving the whole quality of the ERP platform and it should of course be oriented toward improving the performances on all the considered criteria, i.e. on all the metrics related to maintainability and usability aspects. The same consideration is valid for the 1st, 2nd and 3rd scenarios. For all these scenarios, we can further observe as the alternatives corresponding to order scheduling (A_2), shipping (A_3), and invoicing (A_5) fall into the medium-high performance class, whereas the subprogram involving the process of order placement (A_1) has been assigned to the medium-low performance class. Under a practical point of view, this result suggests as this first alternative may be the next in need of improvement upon the fourth alternative, already identified as the main critical aspect of the whole sales program.

When the analyst is instead interested in attributing a strongly pronounced importance to the driver of maintainability with respect to usability, the module to be improved with priority to increase software performance still corresponds to the fourth alternative. However, one can appreciate a difference with respect to the previous considered scenarios. In the fourth scenario, the fifth alternative moves from the medium-high to the medium-low performance class. This means that, after having improved module A_4 related to tracking, either order placement (A_1) or invoicing (A_5) are candidates for improvement. On the contrary, as we can observe in the fifth scenario, the order placement (A_1) is upgraded to the medium-high performance class by the procedure. This depends on the fact that much more importance has been attributed to the driver of maintainability (also to reflect the higher number of its related criteria) with respect to the usability, the last one having associated importance of 15% in the last scenario.

In such a way, it is possible to appreciate that the approach proposed in the present research is capable of highlighting possibilities of improvement in software quality, also by considering the different relative importance attributed to the main drivers of analysis. Once highlighted the module of the ERP platform in major need of improvement, such techniques as Post Factum Analysis (PFA) may be applied to determine the minimal improvement requirements allowing the assignment to a more preferred class. As illustrated by such authors as Kadzinski et al. (2016) and Ciomek et al. (2018) approaches of improvement based on PFA are useful to define future performance targets also in the presence of multiple conflicting criteria. In the case of our case study, ideas of improvement for the order tracking module A_4 may proprietarily regard such aspects as satisfaction, complexity and duplication. In such a way, achieving better performance for both the drivers of usability and maintainability would be possible.

4. Conclusions and future developments

The paper proposes dealing with the problem of ERP modular system categorisation by means of a MCDM approach and, in particular, the ELECTRE TRI technique is suggested to classify software modules based on such drivers as usability and maintainability. To the best of the authors' knowledge, this is the first time that such an approach is carried out, since existing studies in the field barely cover the topic object of the present research paper.

Among the various characteristics to consider for developing a quality software model according to the current international standard, usability and maintainability have been chosen as the main drivers due to their relevance, supported by the existing literature on software engineering. Metrics quantifying these two drivers have been highlighted and related to the various features associated both to usability and maintainability. These metrics constitute the criteria to perform the ELECTRE TRI application.

A case study has been developed with the aim to classify ERP software modules into predefined and ordered classes. Based on the classification results, it is possible to distinguish which modules should be improved with priority to have a positive impact on the whole software performance (modules belonging to the low performance class) and, similarly, which modules may see their modifications postponed (precisely, those modules belonging to high or medium-high performance classes). We believe that the application of the ELECTRE TRI method better suits the problem under analysis with respect to other methods capable to rank alternatives. Indeed, relying on the possibility to sort modules into performance classes instead of ranking ERP modules is more useful to immediately highlight those sets of alternatives in need of improvement. Also, the nature of this need can be easily distinguished on the basis of common characteristics. A final sensitivity analysis on criteria weights was carried out to offer a perspective on results robustness.

Possible future developments of the present research may regard the application of a hybrid MCDM approach aimed at studying interdependencies existing among software quality characteristics and, in general, among the main elements of the problem. Specifically, having been the topic of interdependence among criteria already considered in ELECTRE methods (Figueira et al., 2009), further developments in this direction could regard the implementation of deeper sensitivity analysis by means of Stochastic Multicriteria Acceptability Analysis (Lahdelma et al., 1998; Pelissari et al., 2020), whose applicability in combination with ELECTRE methods has been demonstrated by Corrente et al. (2017). Furthermore, the hierarchical version of ELECTRE TRI methods (Corrente et al., 2016) may be applied to achieve a classification on the basis of relevant ERP macro-criteria in integration with Multiple Criteria Hierarchy Process principles (Corrente et al., 2013).

Uncertainty affecting input data may also be managed by using such mathematical tools as the fuzzy set or the probability theories. We lastly aim to propose PFA-based approaches to specifically determine how to lead the improvement process in practice to assure the upgrade of critical ERP modules towards higher performance classes.

References

Aboabdo, S., Aldhoiena, A., Al-Amrib, H. (2019) Implementing Enterprise Resource Planning ERP System in a Large Construction Company in KSA. Procedia Computer Science, 164, 463–470.

Almeida-Dias, J., Figueira, J. R., Roy, B. (2010) Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions. European Journal of Operational Research, 204(3), 565-580.

Almeida-Dias, J., Figueira, J. R., Roy, B. (2012) A multiple criteria sorting method where each category is characterized by several reference actions: The Electre Tri-nC method. European Journal of Operational Research, 217(3), 567-579.

Al-Rawashdeh, T.A., Alazzeh, F.M., Al-Qatawneh, S.M. (2014) Evaluation of ERP Systems Quality Mode Using Analytic Hierarchy Process (AHP) Technique. Journal of Software Engineering and Applications, 7, 225–232.

Alsolai, H., Roper, M. (2020), A systematic literature review of machine learning techniques for software maintainability prediction. Information and Software Technology, 119, 106214.

Atalag, K., Yang, H.Y., Tempero, E., Warren, J.R. (2014) Evaluation of software maintainability with openEHR – a comparison of architectures. International Journal of Medical Informatics, 83(11), 849–859.

Baggen, R., Correira, J.P., Schill, K., Visser, J. (2012) Standardized code quality benchmarking for software maintainability. Software Quality Control, 20(2), 287–307.

Barak, S., Mokfi, T. (2019) Evaluation and selection of clustering methods using a hybrid group MCDM. Expert Systems with Applications, 138, 112817.

Baykasoğlu, A., Gölcük, İ. (2017) Development of a two-phase structural model for evaluating ERP critical success factors along with a case study. Computers & Industrial Engineering, 106, 256–274.

Bernroider, E.W.N., Stix, T.V. (2013) Profile distance method-a multi-attribute decision making approach for information system investments. Decision Support Systems, 42, 988–999.

Bouwers, E., Deursen, A.V., Visser J. (2014) Towards a catalog format for software metrics. In Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics, ACM, New York, USA, 44–47.

Bødker, S., Sundblad, Y. (2008) Usability and interaction design – new challenges for the Scandinavian tradition. Behaviour & Information Technology, 27(4), 293–300.

Brentan, B., Carpitella, S., Barros, D., Meirelles, G., Certa, A., Izquierdo, J. (2020) Water quality sensor placement: a multi-objective and multi-criteria approach. Water Resources Management, accepted.

Cailloux, O., Meyer, P., Mousseau, V. (2012) Eliciting ELECTRE TRI category limits for a group of decision makers. European Journal of Operational Research, 223(1), 133–140.

Carpitella, S., Ocana-Levario, S.J., Benítez, J., Certa, A., Izquierdo, J. (2018) A hybrid multi-criteria approach to GPR image mining applied to water supply system maintenance. Journal of Applied Geophysics, 159, 754–764.

Certa, A., Carpitella, S., Enea, M., Micale, R. (2016) A multi criteria decision making approach to support the risk management: a case study. Proceedings of the 21th Summer School “Francesco Turco”, Naples, Italy, 242-246.

Ciomek, K., Ferretti, V., Kadziński, M. (2018) Predictive analytics and disused railways requalification: Insights from a Post Factum Analysis perspective. Decision Support Systems, 105, 34-51.

Chang, B., Kuo, C., Wu, C.H., Tzeng, G.H. (2015) Using Fuzzy Analytic Network Process to assess the risks in enterprise resource planning system implementation. Applied Soft Computing, 28, 196–207.

Chen, C.S., Liang, W.Y., Hsu, H.Y., (2015) A cloud computing platform for ERP applications. Applied Soft Computing, 27, 127–136.

Chofreh, A.G., Goni, F.A., Klemeš, J.J., Malik, M.N., Khan, H.H. (2020) Development of guidelines for the implementation of sustainable enterprise resource planning systems. Journal of Cleaner Production, 24420, 118655.

Coleman, D.M., Ash, D., Lowther, B., Oman, P.W. (1994) Using metrics to evaluate software system maintainability. IEEE Computer, 27(8), 44–49.

Corrente, S., Greco, S., Słowiński, R. (2013) Multiple criteria hierarchy process with ELECTRE and PROMETHEE. Omega, 41(5), 820-846.

Corrente, S., Greco, S., Słowiński, R. (2016) Multiple Criteria Hierarchy Process for ELECTRE Tri methods. European Journal of Operational Research, 252(1), 191–203.

Corrente, S., Figueira, J. R., Greco, S., Słowiński, R. (2017). A robust ranking method extending ELECTRE III to hierarchy of interacting criteria, imprecise weights and stochastic analysis. Omega, 73, 1-17.

Costa, C.J., Ferreira, E., Bento, F., Aparicio, M. (2016) Enterprise resource planning adoption and satisfaction determinants. Computers in Human Behavior, 63, 659–671.

Dias, L.C., Mousseau, V. (2006) Inferring Electre's veto-related parameters from outranking examples. European Journal of Operational Research, 170, 172–191.

Doumpos, M., Marinakis, Y., Marinaki, M., Zopounidis, C. (2009) An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELECTRE TRI method. European Journal of Operational Research, 199, 496–505.

Fernández, E., Figueira, J. R., Navarro, J., Roy, B. (2017) ELECTRE TRI-nB: A new multiple criteria ordinal classification method. European Journal of Operational Research, 263(1), 214-224.

Figueira, J. R., Greco, S., Roy, B. (2009) ELECTRE methods with interaction between criteria: An extension of the concordance index. European Journal of Operational Research, 199(2), 478-495.

Figueira, J.R., Greco, S., Roy, B., Slowinski, R. (2010) ELECTRE methods: main features and recent developments. Handbook of multicriteria analysis, Springer, 51–89.

Figueira, J. R., Greco, S., Roy, B., Słowiński, R. (2013) An overview of ELECTRE methods and their recent extensions. Journal of Multi-Criteria Decision Analysis, 20(1-2), 61-85.

Figueira, J.R., Greco, S., Ehrgott, M. (2016). Multiple Criteria Decision Analysis: State of the Art Surveys. Springer, New York.

Garousi, V., Rainer, A., Lauvås, P., Arcuri, A. (2020) Software-testing education: A systematic literature mapping. Journal of Systems and Software, 165, 110570.

Govindan, K., Jepsen, M. B. (2016) ELECTRE: A comprehensive literature review on methodologies and applications. European Journal of Operational Research, 250(1), 1-29.

Greco, S., Figueira, J., Ehrgott, M. (2016) Multiple criteria decision analysis (Vol. 37). New York: Springer.

Gupta, M., Kohli, A. (2006) Enterprise resource planning systems and its implications for operations function. Technovation, 26(5-6), 687–696.

Gupta, P., Pham, H., Mehlawat, M.K., Verma, S. (2013) A Fuzzy Optimization Framework for COTS Products Selection of Modular Software Systems. International Journal of Fuzzy Systems, 15(2), 91–108.

Gupta, S., Chug, A. (2020) Assessing Cross-Project Technique for Software Maintainability Prediction. Procedia Computer, 167, 656–665.

Gürbüz, T., Alptekin, S.E., Alptekin, G.I. (2012) A hybrid MCDM methodology for ERP selection problem with interacting criteria. Decision Support Systems, 54, 206–214.

Hallikainen, P., Kivijärvi, H., Tuominen, M. (2009) Supporting the module sequencing decision in the ERP implementation process - An application of the ANP method. International Journal of Production Economics, 119(2), 259–270.

Hegedűs, P., Kádár, I., Ferenc, R., Gyimóthy, T. (2018) Empirical evaluation of software maintainability based on a manually validated refactoring dataset. Information and Software Technology, 95, 313–327.

Heitlager, I., Tobias, K., Visser, J. (2007) A Practical Model for Measuring Maintainability – a preliminary report. In Proceedings of the 6th International Conference on Quality of Information and Communications Technology, 30–39.

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

ISO/IEC 25010 (2011) Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, https://www.iso.org/standard/35733.html.

Kadzinski, M., Ciomek, K., Rychły, P., Słowiński, R. (2016) Post factum analysis for robust multiple criteria ranking and sorting. Journal of Global Optimization, 65(3),1-32.

Karsak, E.E., Özogul, C.O. (2009) An integrated decision making approach for ERP system selection. Expert Systems with Applications, 36,660–667.

Kilic, H.S., Zaim, S., Delen, D. (2015) Selecting "The Best" ERP system for SMEs using a combination of ANP and PROMETHEE methods. Expert Systems with Applications, 42(51), 2343–2352.

Kortum, P.T., Bangor, A. (2013) Usability ratings for everyday products measured with the System Usability Scale. International Journal of Human-Computer, 29(2), 67–76.

Lahdelma, R., Hokkanen, J., Salminen, P. (1998) SMAA-stochastic multiobjective acceptability analysis. European Journal of Operational Research, 106(1), 137-143.

Lee, H.-Y., Wang, N.-J. (2019) Cloud-based enterprise resource planning with elastic model–view–controller architecture for Internet realization. Computer Standards & Interfaces, 64, 11–23.

Lee, J.Y., Kim, J.Y., You, S.J., Kim, Y.S., Koo, H.Y., Kim, J.H., Kim, S., Park, J.H., Han, J.S., Kil, S., Kim, H. (2019) Development and Usability of a Life-Logging Behavior Monitoring Application for Obese Patients. Journal of Obesity & Metabolic Syndrome, 28(3), 194–202.

Lin, C.T., Chen, C.B., Ting, Y.C. (2011) An ERP model for supplier selection in electronics industry. Expert Systems with Applications, 38, 1760–1765.

Liu, Z., Ming, X. (2019) A methodological framework with rough-entropy-ELECTRE TRI to classify failure modes for co-implementation of smart PSS. Advanced Engineering Informatics, 42, 100968.

López, C., Salmeron, J.L. (2012) Monitoring software maintenance project risks. Spain Procedia Technology, 5, 363–368.

Lourenço, R., Costa, J.P. (2004) Using ELECTRE TRI outranking method to sort MOMILP non dominated solutions. European Journal of Operational Research, 153, 271–289.

Malekmohammadi, B., Zahraie, B., Kerachian, R. (2011) Ranking solutions of multi-objective reservoir operation optimization models using multi-criteria decision analysis. Expert Systems with Applications, 38, 7851–7863.

Merad, M., Verdel, T., Roy, B., Kouniali, S. (2004) Use of multi-criteria decision-aids forrisk zoning and management of large area subjected to mining-induced hazards. Tunnelling and Underground Space Technology, 19(2), 125–138.

Mousseau, V., Słowiński, R. (1998) Inferring an ELECTRE TRI model from assignment examples. Journal of Global Optimization, 12(2), 157–174.

Mousseau, V., Słowiński, R., Zielniewicz, P. (2000) A user-oriented implementation of the ELECTRE TRI method integrating preference elicitation support. Computers & Operations Research, 27(7-8), 757–777.

Mousseau, V., Figueira, J., Naux, J.-Ph. (2001) Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results. European Journal of Operational Research, 130(2), 263-275.

Mousseau, V., Ngo The, A. (2002) Using assignment examples to infer category limits for the ELECTRE TRI method. Journal of Multi-Criteria Decision Analysis, 11(1), 29–43.

Nofal, M.I., Yusof, Z.M. (2013) Integration of Business Intelligence and Enterprise Resource Planning within Organizations. Procedia Technology, 11, 658–665.

Parthasarathy, S., Sharma, S. (2016) Efficiency analysis of ERP packages - A customization perspective. Computers in Industry, 82, 19–27.

Pedroli, E., Greci, L., Colombo, D., Serino, S., Cipresso, P., Arlati, S., Mondellini, M., Boilini, L., Giussani, V., Goulene, K., Agostoni, M., Sacco, M., Stramba-Badiale, M., Riva, G., Gaggioli, A.

(2018) Characteristics, Usability, and Users Experience of a System Combining Cognitive and Physical Therapy in a Virtual Environment: Positive Bike. Sensors, 18, 2343.

Pelissari, R., Oliveira, M. C., Amor, S. B., Kandakoglu, A., Helleno, A. L. (2020) SMAA methods and their applications: a literature review and future research directions. Annals of Operations Research, 293, 433-493.

Ramezanian, R. (2019) Estimation of the profiles in posteriori ELECTRE TRI: A mathematical programming model. Computers & Industrial Engineering, 128, 47–59.

Raza, A., Fernando Capretz, L., Ahmed, F. (2011) An Empirical Study of Open Source Software Usability: The Industrial Perspective. International Journal of Open Source Software and Processes, 3(1), 1-16.

Rodríguez, R., Molina-Castillo, F.-J., Svensson, G. (2020) The mediating role of organizational complexity between enterprise resource planning and business model innovation. Industrial Marketing Management, 84, 328-341.

Roy, B. (1991) The outranking approach and the foundations of ELECTRE methods. Theory and Decision, 31(1), 49–73.

Roy, B., Bouyssou, D. (1993) Aide Multicritère à la Décision: Méthodes et Cas, Economica, Paris, 1993.

Roy, B., Figueira, J. R., Almeida-Dias, J. (2014) Discriminating thresholds as a tool to cope with imperfect knowledge in multiple criteria decision aiding: Theoretical results and practical issues. Omega, 43, 9-20.

Ruivo, P., Johansson, B., Sarker, S., Oliveira, T. (2020) The relationship between ERP capabilities, use, and value. Computers in Industry, 117, 103209.

Saaty, T. L. (1980) The Analytic Hierarchy Process. New York, McGraw-Hill.

Sánchez-Lozano, J.M., García-Cascales, M.S., Lamata, M.T. (2016) Comparative TOPSIS-ELECTRE TRI methods for optimal sites for photovoltaic solar farms. Case study in Spain. Journal of Cleaner Production, 127, 387–398.

Saraiva, J., De França, M.S., Soares, S.C.B., Filho, F.J.C.L., De Souza, R.M.C.R. (2015) Classifying metrics for assessing Object-Oriented Software Maintainability: A family of metrics' catalogs. Journal of Systems and Software, 103, 85–101.

Sauro, J., Kindlund, E. (2005) A method to standardize usability metrics into a single score. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 401–409.

Siskos, Y., Grigoroudis, E., Krassadaki, E., Matsatsinis, N. (2007) A multicriteria accreditation system for information technology skills and qualifications. European Journal of Operational Research, 182, 867–885.

Vogel-Heuser, B., Ocker, F. (2018) Maintainability and evolvability of control software in machine and plant manufacturing — An industrial survey. Control Engineering Practice, 80, 157–173.

Winter, S., Wagner, S., Deissenboeck, F. (2007) A Comprehensive Model of Usability. Engineering Interactive Systems, 4940, 106–122.

Xu, L. (2012) The Evaluation of ERP Sandtable Simulation Based on AHP. Physics Procedia, 33, 1924–1931.

Yazgan, H.R., Boran, S., Goztepe, K. (2009) An ERP software selection process with using artificial neural network based on analytic network process approach. Expert Systems with Application, 36, 9214–9222.

Yu, W. (1992) Aide multicritère à la décision dans le cadre de la problématique du tri: méthodes et applications. PhDthesis, LAMSADE, Universit'e Paris Dauphine, Paris.