Information and Software Technology 51 (2009) 555–563

Contents lists available at ScienceDirect

Information and Software Technology

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

Evaluating and selecting software packages: A review

Anil S. Jadhav [a,*], Rajendra M. Sonar [b]

[a] Computer Department, Sinhgad Institute of Management, Wadgaon (Bk), Pune 411 041 India
[b] Shailesh J. Mehta School of Management, Indian Institute of Technology, Powai, Mumbai 400 076, India

ARTICLE INFO

Article history:
Received 6 March 2008
Received in revised form 14 July 2008
Accepted 4 September 2008
Available online 30 October 2008

Keywords:
Software evaluation
Software selection
Evaluation criteria
Software selection tools

ABSTRACT

Evaluating and selecting software packages that meet an organization's requirements is a difficult software engineering process. Selection of a wrong software package can turn out to be costly and adversely affect business processes. The aim of this paper is to provide a basis to improve the process of evaluation and selection of the software packages. This paper reports a systematic review of papers published in journals and conference proceedings. The review investigates methodologies for selecting software packages, software evaluation techniques, software evaluation criteria, and systems that support decision makers in evaluating software packages. The key findings of the review are: (1) analytic hierarchy process has been widely used for evaluation of the software packages, (2) there is lack of a common list of generic software evaluation criteria and its meaning, and (3) there is need to develop a framework comprising of software selection methodology, evaluation technique, evaluation criteria, and system to assist decision makers in software selection.

Contents

* Corresponding author. Tel.: +91 020 4356592; fax: +91 020 24356592.
E-mail addresses: a_s_jadhav74@yahoo.co.in (A.S. Jadhav), rm_sonar@iitb.ac.in (R.M. Sonar).

0950-5849/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.infsof.2008.09.003

1. Introduction

In the past few years there has been increase in the demand for computer software packages. Software firms have produced a variety of packages in response to this demand. Software packages provide a large number of features that are customizable and can be tailored to meet the specific needs of the organizations. Improper selection of a software package may result in wrong strategic decisions with subsequent economic loss to the organization. For example, there are a number of solutions in an ERP market and every solution has different features. As ERP packages cost hundreds of thousands and even millions of dollars, purchasing an ERP solution is a high expenditure activity that consumes a significant portion of companies' capital budgets [69]. Selecting the right solution is an exhausting process for companies [18]. Therefore, selecting a software package that meets the requirements needs a full examination of many conflicting factors and it is a difficult task. This has led researchers to investigate better ways of evaluating and selecting software packages. The purpose of this paper is to review the research work done in the field of evaluating and selecting software packages and provide a basis to improve process of the software selection. Keeping this objective in mind, the scope of review is limited to the literatures that suggest criteria for software selection, methodologies for software selection, software evaluation techniques and systems/tools to assist decision makers in evaluating and selecting software packages. In this paper we address the following research questions:

- RQ1: What is the contribution of the literature in the field of evaluation and selection of the software packages?
- RQ2: What are the methodologies for selecting software packages? This question leads to another sub-question: What are the stages in the software selection methodology?
- RQ3: What are the systems/tools to assist decision makers in evaluating and selecting software packages?
- RQ4: What are the software evaluation techniques?
- RQ5: What are the software evaluation criteria?

Software evaluation can be formulated as multiple criteria decision making (MCDM) problem. MCDM refers to making preference decisions over the available alternatives that are characterized by multiple, usually conflicting, attributes [68,75]. The goal of the MCDM is [41]:

- to help decision makers choose the best alternative of those studied
- to help sort out alternatives that seem good among the set of alternatives studied
- to help rank the alternatives in decreasing order of performance.

In recent years, researchers have focused on models and methods for reusable off-the-shelf software selection [5,13,17,29,31,32,34,36,48,59]. However, there exists other literature that:

- concentrate on evaluation and selection of specific software products such as CASE tools [6,38,53], simulation software [10,19,44–46,66], DSS software [7,55], AHP software [49], knowledge management tools [47,50], data mining software [11], visual programming languages [26], ERP packages [21], CRM packages [12], expert system shells [64], and operations management software [61]
- describe automated systems/tools that assist decision makers in various activities involved in software evaluation and selection [4,17,20,23,39,70]
- describe only criteria for software selection [3,9,54,55,64], and methodology for software selection [2,6,19]
- relate to the evaluation of a single software attribute, quality or some quality sub-attribute, for a software product [15,16].

Stamelos and Tsoukias [62] analyzed the contents of different "problem situations" and suggested a basic classification of software evaluation problem situations: keep or change; make or buy; commercial product evaluation; tender evaluation; software certification; software process evaluation; software system design selection.

The remainder of this paper is organized as follows. Section 2 describes the research method applied in this review. Results are presented in Section 3. The paper is concluded in Section 4.

2. Research method

2.1. Inclusion criteria

The main criterion used for including a paper in our review is that paper should describe research in the field of evaluating and selecting software packages. Only papers that describe: (i) methodology for selecting software packages, and/or (ii) software evaluation criteria, and/or (iii) software evaluation technique, and/or (iv) system/tool to assist decision makers in evaluating software packages, are included in our review. We exclude pure discussion or opinion papers and papers that describe evaluation technique in general and not applied to software evaluation. There were examples of papers describing the same study in more than one journal paper. Fortunately, the number of such cases was small and would not lead to important changes in the outcome of our analysis. Therefore we decided not to exclude any papers for that reason.

2.2. Search strategy, and search

The search strategy for the review is directed towards finding published papers in archival journals, conference proceedings and technical reports from the contents of four electronic databases namely, ACM portal, Elsevier's Science Direct, IEEE Xplore, and Springer-Verlag's Link. The search terms used were "software selection criteria", "software evaluation techniques", "software selection methodologies", "evaluating and selecting software packages", "method for evaluating and selecting software packages", "criteria for evaluating and selecting software packages", "software evaluation criteria", "systems/tools for evaluation and selection of software packages", "knowledge-based systems for software selection", "framework for evaluating and selecting software packages", and "software selection process". Other relevant journals we found while searching the articles on this topic are "information and management", "Information and software technology", and "European journal of operational research". Articles published in proceedings of IEEE on Software Engineering, Springer-Verlag, International conference on COTS-Based software system are also found relevant to this topic. The series of articles on evaluating software engineering methods and tools, part 5 to part 8, ACM SIGSOFT, is one of the major contributions to this topic.

2.3. Paper selection

Our selection process had two parts: (i) an initial selection from the search results, based on reading the abstract of the papers, and (ii) final selection from the initially selected list of papers, based on reading of entire paper. The initial list consists of 130 papers which we found relevant to the topic and potential candidates for inclusion in our review. Initial selection of the paper was done jointly by both the authors on the basis of reading title and abstract of the paper. The first author of the paper then read all 130 papers

in detail and considered 64 papers to be included in the final list for review. In the second phase of paper selection we eliminated 4 papers that did not give any useful information on evaluation criteria, evaluation technique, selection methodology, and systems/tools for software selection. The second author of the paper, cross checked whether papers in the final list considered for review addressed the research question and contributed to the basic purpose of the review. A random sample of 20 papers were selected for the cross checking. There was no disagreement on final selection of papers. The search began in early 2006 and completed in early 2007.

2.4. Data extraction

In the data extraction phase, the first author of this paper read every selected paper and extracted information about the attributes defined in Table 1. The extracted data were then cross checked by second author of this paper by random selection of 20 papers i.e. about 30% of the total. During data extraction phase we found that four papers did not give any useful information on software selection methodology, evaluation criteria, evaluation technique, and system/tool for software selection, therefore those papers are not considered while presenting results of the review.

3. Results

This section describes analysis of the data extracted from our selected studies. The contribution of the reviewed literature in the field of evaluation and selection of the software packages is presented in Section 3.1. Section 3.2 presents stage-based methodologies for selecting software packages. Section 3.3 describes systems/tools to assist decision makers in evaluating software packages. Techniques for evaluating software packages and software evaluation criteria are described in Sections 3.4 and 3.5, respectively. Section 3.4.2 discusses the limitations of the study.

3.1. Contribution of the literature in the field of evaluation and selection of the software packages (RQ1)

We have classified different levels of contribution of reviewed literature into four categories: methodologies for software selection, software evaluation techniques, software evaluation criteria, and systems/tools to support decision makers in software selection. A summary of the contribution of the reviewed literature in the field of evaluation and selection of software packages is given in Table 2. Column 1 is the name of the author(s). Column 2 is the specific type of software considered for evaluation/selection. Column 3, 4 and 5 shows whether literature has suggested criteria for evaluation, stage-based methodology for software selection, and evaluation technique for evaluation of software given in column three, respectively. Columns 6 indicate whether the proposed selection methodology, evaluation technique, selection criteria have been applied practically. Only 6 papers [4,17,20,23,39,70] describes a system/tool for software selection, hence reference of these papers and column representing a system/tool is not included Table 2. The contribution of the reviewed literature in each category is represented graphically in Fig. 1.

3.2. Software selection methodologies (RQ2)

Methodologies illustrate the factors and issues that should be taken into consideration during selection of the software packages. The methodology is not intended as rigid structure that must be followed without any deviation, it is intended as guideline and an aid that can be adapted according to the requirements of the individual organization [50]. A stage-based methodology for selecting software packages was described in 27 different studies, out of which 7 studies concentrate on COTS component selection and remaining concentrate on selection of specific type of the software package.

On the basis of review of literature we propose a generic stage-based methodology for selection of any software package which consists of following seven stages.

1. Determining the need for purchasing the system and preliminary investigation of the availability of packaged software that might be suitable candidate, including high level investigation of software features and capabilities provided by vendor [6,7,12,31,50,64,74].
2. Short listing of candidate packages [5–7,12,21,31,50,66].
3. Eliminating most candidate package that do not have required feature or do not work with the existing hardware, operating system and database management software or network [5–7,21,31,50].
4. Using an evaluation technique to evaluate remaining packages and obtain a score or overall ranking of them [5–7,31,48,64,74].
5. Doing further scrutiny by obtaining trial copy of top software packages and conducting an empirical evaluation. Pilot testing the tool in an appropriate environment [21,50].
6. Negotiating a contract specifying software price, number of licenses, payment schedule, functional specification, repair and maintenance responsibilities, time table for delivery, and options to terminate any agreement [12,21].
7. Purchasing and implementing most appropriate software package.

We found that none of the primary studies explicitly covered step 7. However, good evaluation practice suggests that some action should be taken to ensure that the selected package performs as well as expected after implementation.

3.3. Systems/tools for evaluation and selection of software packages (RQ3)

The process of evaluating and selecting software packages involves different activities as defined in section 3.2. The need of having a system/tool that supports these activities arose. Some sys-

Table 1
Data extracted from each study

Attribute	Description
Reference number	This indicates reference number of the paper considered for study
Authors	The authors of the paper
Year	The Year of the publication
Title	The title of the publication
Software	Does the paper apply to a particular type of software? If Yes: What type of the software?
Criteria	Does the paper describe criteria used to assess the packages? If Yes: What are the criteria?
Selection methodology	Does the paper define methodology for software package selection? If Yes: What are the stages in software selection methodology?
Evaluation technique	Does the paper include evaluation technique? If Yes: What is evaluation technique?
System/tool	Does the paper specify any systems/tools to support software package Selection? If Yes: What are basic functions of the system/tool?

Table 2
Summary of contribution of the reviewed literature

Author(s)	Software type	Criteria	Selection methodology	Evaluation technique	Practical application
Adhikari et al. [1]	Accounting software	Yes	No	No	No
Arditi and Singh [3]	Accounting software	Yes	No	No	No
Bhuta and Boehm [5]	COTS components	No	Yes	No	Yes
Blanc and Korn [6]	CASE tools	Yes	Yes	No	No
Blanc and Jelassi [7]	DSS software	Yes	Yes	Yes	Yes
Chau [9]	Software in small businesses	Yes	No	No	No
Cochran and Chen [10]	Simulation software	Yes	No	Yes	Yes
Collier et al. [11]	Data mining software	Yes	Yes	Yes	Yes
Colombo et al. [12]	CRM packages	Yes	Yes	Yes	Yes
Comella-Dorda [13]	COTS products	No	Yes	No	No
Davis and Williams [14]	Simulation software	Yes	No	Yes	Yes
Franch and Carvallo[15,16]	Software packages	Yes	No	No	Yes
Hlupic and Paul [19]	Simulation software	Yes	Yes	No	Yes
Illa et al. [21]	ERP	Yes	Yes	No	No
Jarkee and Vassiliou [22]	Database query language	Yes	No	Yes	Yes
Kim and Yoon [25]	Expert system shell	Yes	No	Yes	Yes
Kiper et al. [26]	Visual programming language	Yes	No	Yes	Yes
Kitchenham [27]	SE methods and tools	Yes	Yes	Yes	Yes
Kontio [30,31]	COTS selection	Yes	Yes	Yes	Yes
Kunda [32]	COTS components	Yes	Yes	Yes	Yes
Lai et al. [33]	Multimedia authoring system	Yes	No	Yes	Yes
Lawlis et al. [34]	COTS Software products	Yes	Yes	Yes	Yes
Lee et al. [35]	Software selection	No	No	Yes	No
Leuing et al. [36]	COTS product	No	Yes	No	No
Lin et al. [37]	Data warehouse system	Yes	Yes	Yes	Yes
Misra [38]	CASE tools	Yes	No	No	No
Morera [42]	COTS product	No	Yes	Yes	No
Morisio and Tsoukias [43]	Software products	Yes	Yes	Yes	Yes
Nikoukaran et al. [45]	Simulation software	Yes	No	No	No
Ngai and Chan [47]	Knowledge management tools	Yes	No	Yes	Yes
Oh and Lee [48]	COTS components	Yes	Yes	Yes	Yes
Ossadnik [49]	AHP software	Yes	No	Yes	Yes
Patel and Hlupic [50]	Knowledge management tool	Yes	Yes	No	No
Perez and Rojas [51]	Workflow type software	Yes	No	Yes	Yes
Phillips-Wren et al. [52]	Decision support systems	Yes	Yes	Yes	Yes
Plessis [53]	CASE tool	Yes	No	Yes	No
Reed [54]	GIS software	Yes	No	No	No
Reimann and Waren [55]	DSS software	Yes	No	No	No
Rincon et al. [56]	Simulation software	Yes	Yes	No	Yes
Sanders et al. [58]	Computer software packages	Yes	No	Yes	Yes
Santiago et al. [59]	COTS software product	No	Yes	No	No
Sarkis and Talluri [60]	e-commerce software	Yes	No	Yes	Yes
Shtub et al. [61]	Operations management	Yes	No	Yes	Yes
Stylianou [64]	Expert system shell	Yes	Yes	No	No
Teltumbde [65]	ERP Projects	Yes	Yes	Yes	Yes
Tewoldeberhan et al. [66]	Discrete-event simulation	Yes	Yes	Yes	Yes
Toshtzar [67]	Computer software	Yes	No	Yes	Yes
Wei et al. [71]	ERP system	Yes	Yes	Yes	Yes
Welzel and Hausen [72]	Software product	Yes	Yes	No	No
Wit and Herroelen, [73]	Project management software	Yes	No	No	Yes
Wright [74]	Instructional support system	Yes	Yes	Yes	No
Zahedi [77]	Expert systems	Yes	No	Yes	Yes

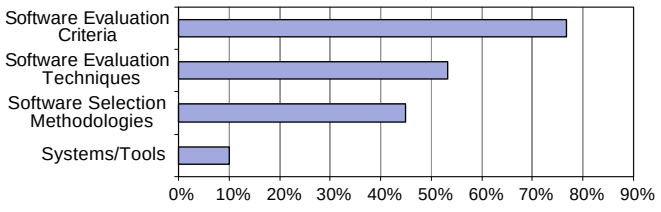

Fig. 1. Contribution of the reviewed literature.

tems/tools already exist to partially support these activities. We found six papers describing system/tool to assist decision makers in evaluating and selecting software packages selection [4,17,20,23,39,70].

Kathuria et al. [23] presented a knowledge-based system that is designed to assist managers in selecting IT applications that are consistent with competitive priorities and process structure of a manufacturing company. The system was developed using 1st class KBS shell that uses ID3 induction algorithm. Validation of the system illustrated that its performance is consistent with the human experts, and it has potential to facilitate and swift decision making in the selection of appropriate IT applications that best match an organization's manufacturing strategy.

SimSelect is another tool that assists users in the simulation software selection process. SimSelect consists of a database that holds the information related to evaluation details of each package, which is linked to an interface developed using VB 3.0. The system queries a database and finds a simulation package suitable to the user, based on requirements which have been specified. It also provides a recommendation of alternative packages suitable to the user, and allows prioritization of requirements in levels of importance [20].

Vlahavas et al. [70] presented ESSE (Expert System for software evaluation), a prototype expert system for software evaluation that embodies various aspects of the multi-criteria decision aid methodology. Main features of the system are: flexibility in the problem

modeling, built in knowledge about software problem solving and software attribute assessment. The system allows evaluators to define their own attributes, along with their measurement definitions. Evaluation problems are modeled around software attributes such as quality and cost. Expert assistance guides the evaluator in feeding values to the decision model.

A software system called DesCOTS (description, evaluation and selection of COTS component) developed by Grau [17] includes various tools to support the COTS component selection process: the quality model tool allows defining quality models; the COTS evaluation tool allows the evaluation of components; the COTS selection tool allows the definition of requirements that drive the COTS component selection; and the taxonomy tool allows organizing the COTS domain as a taxonomy supporting reuse of quality models.

Bandini et al. [4] presented a knowledge-based tool developed in order to support business managers, software architects, and engineers in the design process of the COTS-based system solutions. The tool has been designed and developed based on expert's knowledge and experience. Basically, the tool addresses the problem of selecting, integrating and deploying COTS components to deliver tailored software systems.

Mohamed et al. [39] proposed a conceptual model to support decision makers in COTS selection process. The model can be implemented by a system that uses agent technologies supported by two kinds of knowledge bases: a COTS knowledge base (CKB) and a method knowledge base (MKB). The CKB stores information quantitatively about different COTS candidates. MKB stores knowledge required to use properly different methods and techniques during the evaluation and negotiation processes. The main features of the system are: (a) it uses two knowledge bases, CKB and MKB, to help continuous accumulating, managing and reusing of relevant knowledge; and (b) it employs agent technology to facilitate negotiations between different stakeholders and provide them with a quick alternative scenario to select from.

We found that none of the existing systems/tools, referred in this study, support all stages in the software selection methodology as they were not designed and developed with the intention to do so. But each one of them supports some of the activities that need to be carried out during evaluation of the software packages. Table 3 provides summery view of evaluation activities supported by the systems/tools.

In our opinion, expert system for software evaluation [70] appears to be a good choice for a general purpose software selection tool since it supports many of the important evaluation activities.

3.4. Software evaluation techniques (RQ4)

Software evaluation is multi-criteria decision making problem that refers to making preference decisions over the available alternatives. We found that AHP has been widely used for evaluation of the software packages. AHP was developed by Saaty [57] and has been identified as an important approach to multi-criteria decision making problems of choice and prioritization. AHP is based on a hierarchical framework of criteria. The upper level deals with the goal of the selection process. The next level defines the major factors which are subdivided into their constituents in lower levels of hierarchy. The bottom level contains the alternatives to be analyzed. Local priorities are established for each factor on a given level with respect to each factor on the level immediately above it. This is done by pair-wise comparison between the factors at each level. If N factors are being compared then $N(N-1)/2$ pair-wise comparisons are made. These comparisons are the basis for calculation of the relative weight of each factor at each level. The last step of the analysis consists of computing the relative score of each alternative with respect to the decision making goal. The application of AHP to the evaluation of software package has been successfully applied in many research studies [12,14,24,25,30,31,33,40,42,47,49,52,60,61,65,67,71,76,77].

Another technique used for evaluation of software package is the weighted scoring method [7,11,51]. In this method weights and rating scales are assigned to each criterion. The weight reflects the relative importance of each of the criteria while the rating scale indicates how easily each package is able to meet the specific criterion. The rating scales are then multiplied by weight factor of each criterion. Using this scheme a score is calculated for every criterion for each tool. These scores are then totaled to produce a score for each criteria category. Finally, the categorical scores are combined to calculate an overall tool score. A similar approach is used by Kitchenham [28] in her series of article on evaluating software engineering methods and tools, part 5–8, ACM SIGSOFT, to perform specific type of evaluation exercise called feature analysis.

A fuzzy based approach for software evaluation has been used in four different studies [8,10,35,37]. This technique is used when performance rating and weights can not be given precisely. In such cases the fuzzy set theory is used to model the uncertainty of human judgments and such problem is known as fuzzy multiple criteria decision making (FMCDM).

Different techniques for evaluating software packages have been proposed in the literature. Here we attempt to assess the

Table 3
Summary of evaluation activities supported by the system/tool

System/tool	Evaluation activities supported
Knowledge-based system for selecting IT applications [23]	– Defining requirements of the software package in terms of evaluation criteria – Assigning weight (importance) to each evaluation criteria – Recommending software package/s that is consistent with the user requirements
SimSelect: a system for simulation software selection [20]	– Defining requirements of the desired software in terms of evaluation criteria – Prioritizing requirements in terms of level of its importance in the selection process – Recommending software suitable to the user
Expert system for software evaluation [70]	– Defining type of evaluation problem e.g. sorting, classification – Defining evaluation criteria – Defining scales and measures associated with each evaluation criteria – Setting the importance of each evaluation criteria in the selection process – Selecting an appropriate evaluation method – Ranking the alternative software packages – Allowing user to give feedback on the evaluation result to facilitate future reuse of the evaluations
DesCOTS: A software system for selecting COTS components [17]	– Defining an hierarchy of quality factors and their associated metrics – Defining requirements of the COTS component using the quality model – Evaluating and selecting candidate COTS components using the quality model
Knowledge-based tool [4]	– Defining desired functionalities of the component – Identifying and selecting components that implements given functionalities
DSS for COTS selection [39]	– Defining desired functionalities and their preferences – Ranking the available alternatives

strength and weaknesses of evaluation techniques proposed in the literature. This assessment is based on analysis of practical applications of the evaluations techniques discussed in the reviewed papers.

3.4.1. Analytic hierarchy process

Strengths:

- AHP enables decision makers to structure a decision making problem into a hierarchy, helping them to understand and simplify the problem.
- It is flexible and powerful tool for handling both qualitative and quantitative multi-criteria problems.
- AHP procedures are applicable to individual and group decision making.

Weaknesses:

- AHP is time consuming because of the mathematical calculations and number of pair-wise comparisons that increases as the number of alternatives and criteria increases.
- The decision makers need to re-evaluate alternatives when the number of criteria or alternatives are changed.
- Ranking of alternatives depends on the alternatives considered for evaluation hence adding or deleting alternatives can lead to changes in the final rank.

3.4.2. Feature analysis

Strengths:

- Evaluation can be done to any required level of detail by organizing evaluation in different ways such as screening mode, case study, formal experiment and survey.
- It is used not only for technical evaluation but also for evaluation of viability of supplier.

Weaknesses:

- Producing the single number from the individual scores may be misleading because many different combinations of numbers can produce the same aggregate score.

3.4.3. Weighted average sum (WAS)

Strengths:

- Main advantage of WAS is its ease of use.

Weaknesses:

- Weights to the attribute are assigned arbitrary and it is very difficult to assign weight when number of criteria is high.
- To obtain a score using this method a common numerical scaling is required.
- Difficulties emerge when WAS is applied to multi-dimensional MCDM problems.

3.4.4. Fuzzy based approach

Strengths:

- The decision makers can use linguistic terms to evaluate alternatives easily and intuitively.
- It improves decision making procedure by accommodating the vagueness and ambiguity occurred during human decision making.

Weaknesses:

- Difficult to compute fuzzy appropriateness index values and ranking values for all alternatives.

3.5. Evaluation criteria (RQ5)

Even though there are few papers describing systems/tools for software package selection, many of them provide a hierarchy of criteria for evaluation of a specific software package. However, we found no paper describing a generic list of the criteria that can be used for evaluation of any software packages. Criteria to be considered for software evaluation are usually classified in several groups. Quality characteristics of the software package such as functionality, reliability, usability, efficiency, maintainability, and portability have been used as evaluation criteria group in several studies [15,16,27,43,48,49,56,63,72]. Among the ISO/IEC standards related to software quality, ISO/IEC 9126-1 specifically addresses quality model definition and its use as framework for software evaluation. Criteria related to: (1) vendor, (2) hardware and software requirements, and (3) cost and benefits of the software packages are commonly used across many papers. However, criteria related to the output characteristics of the software packages are discussed only in three papers and criteria related to the opinion about the software package are discussed only in one paper.

We have classified the criteria discussed in the literature, using the quality model approach suggested by Franch and Carvallo [15]. The meaning of each criterion is also defined and so the criteria can be used as generic software evaluation criteria. Criteria related to quality of the software are given in Tables 4 and 5. Other criteria related to vendor, cost and benefit, hardware and software, output, and opinions are presented in separate tables from Tables 6–10.

3.6. Limitations of the study

This study has the usual limitations associated with any systematic literature review. In particular, the search process

Table 4
Criteria related to functional characteristics of the software package

Criteria	Criteria group	Criteria meaning
Included functionality	Functional	Areas or functions of the company that the software has to serve. It describes how software package covers each function
Main target	Functional	Functional area(s) for which software is specially oriented or strong
Completeness	Functional	It is defined as degree to which software satisfies functional requirements
Adaptability	Functional	Possible level of customization in general and for the specific company
Openness	Functional	Level of openness to additional development (internal and external) and to other existing applications
Interoperability	Functional	Capability to integrate with other tools and applications
Security levels	Functional	Breadth of security policies supported by the software package (user identification, auditing, data encryption)
Number of simultaneous users	Functional	Number of simultaneous users that can be linked and served by the system

Table 5
Criteria related to software quality characteristics

Criteria	Criteria group	Criteria meaning
Vertical solutions	Personalizability	Number of customized versions of a package accommodating the typical requirements of a specific industry
Customizable fields	Personalizability	Ability to personalize the layout of package interface
Customizable reports	Personalizability	Ability to personalize the layout of reports produced by package
Interface type	Personalizability	Interface type of the package
Programming languages	Personalizability	Ability to personalize modules by programming languages
Middleware standard	Portability	Breadth of middleware standard that are supported by software package (CORBA, DCOM, RMI, ODBC, JDBC, OLE-DB)
DBMS standards	Portability	Breadth of database management systems that can be accessed by software package (SQL server, oracle, DB2, Sybase, Informics)
Communication standards	Portability	Inter-organizational data exchange standards that are supported by software package (EDI, XML)
Platform variety	Portability	Capability of the software package to run on wide variety of computer platforms
Number of modules	Maintainability	Average size of independent code units
Number of independently installable modules	Maintainability	Level of independence among modules
Number of workstations	Maintainability	Maximum number of users that can be supported
Maximum number of distribution tiers	Maintainability	Ability to split package into separate application that can be distributed onto different servers
Number of modules that can be installed on separate servers	Maintainability	Ability to distribute modules on different servers
Scalability	Maintainability	Ability of the software package to handle increasing number of users and higher load of transaction
User interface	Usability	Ease with which user can use interface of the software package
User types	Usability	Ability of the software package to support beginners, intermediate, and advanced users or a combination of user types
Data visualization	Usability	Capability of the software package to present data effectively
Error reporting	Usability	Error reporting and messaging ability of the software package
Domain variety	Usability	Capability of the software package to be used in different industries to solve different kinds of business problems
Ease of use	Usability	Ease with which user can learn and operate the software package
Robustness	Reliability	Capability of the software package to run consistently without crashing
Backup and recovery	Reliability	Capability of the software package to support backup and recovery feature
Time behavior	Efficiency	Ability of the software package to produce results in reasonable amount of time relative to data size

Table 6
Criteria related to the vendor

Criteria	Criteria group	Criteria meaning
User manual	Vendor	Availability of user manual with indexes, with important information and the main commands
Tutorial	Vendor	Availability of tutorial to learn how to use the software package
Troubleshooting guide	Vendor	Availability of troubleshooting guide
Training	Vendor	Availability of training courses to learn the package
Maintenance and upgrading	Vendor	Vendor support for upgrading and maintenance of the software
Consultancy	Vendor	Availability of technical support and consultancy by the vendor
Communication	Vendor	Communication between user and vendor
Demo	Vendor	Availability of on-site demo and free-trial version
Number of installations	Vendor	Number of installations of the software package
Response time	Vendor	Level of service rendered by the vendor
Length of experience	Vendor	Experience of vendor about development of the software product
Product history	Vendor	Popularity of vendor product in the market
Vendor popularity	Vendor	Popularity of vendor in the market
Technical and business skills	Vendor	Technical and business skills of the vendor
Past business experience	Vendor	Past business experience with the vendor, if any
References	Vendor	Number of references of the existing customers using the product

Table 7
Criteria related to cost and benefits

Criteria	Criteria group	Criteria meaning
License cost	Cost	License cost of the product in terms of number of users
Training cost	Cost	Cost of training to the users of the system
Installation and implementation cost	Cost	Cost of installation and implementation of the product
Maintenance cost	Cost	Maintenance cost of the product
Upgrading cost	Cost	Cost of upgrading of the product when new version will be launched
Cost of hardware	Cost	Cost of machinery used to support the system, including processor, memory and terminals
Direct benefits	Benefits	Tangible savings in labor and equipment, reduction in processing cost per unit and elimination of outside service charges
Indirect benefits	Benefits	Improvement in customer service, faster turnaround time of processing

may have missed some relevant papers and the data extraction may have been incorrect for some papers. With respect to the search process, we have limited ourselves to English language studies and to four major electronic databases and search terms related to the terms ‘‘evaluation” and ‘‘selection”. This strategy will not find non-English language papers, paper in many national (as opposed to international) journals and conferences, or papers that use unusual terminology. Overall, we would not expect to have missed a large number of important studies.

Table 8
Criteria related to hardware and software

Criteria	Criteria group	Criteria meaning
Internal memory	Hardware	Primary storage needed to run the software package
Communication protocols	Hardware	Communication protocols supported by the package
External storage	Hardware	Secondary storage needed in the form of disk space and other storage facilities
Compatibility	Software	Compatibility with the existing software and hardware
Source code	Software	Availability of the source code
Hardware platform	Hardware	Hardware platform required to run the software
Network configuration	Configuration	Network technology needed to run the software package e.g. LAN, WAN

Table 9
Criteria related to the opinions from technical and non-technical sources

Criteria	Criteria group	Criteria meaning
Opinions-technical sources	Opinions	Opinions about the software package from – Potential vendors/sales representatives – In-house experts – External consultants – Computer/IS trade magazines, software product leaflets
Opinions-non technical sources	Opinions	Opinions about software package from – Subordinates – End users – Outside personal acquaintances

Table 10
Criteria related to the output

Criteria	Criteria group	Criteria meaning
Report	Output	Standard and customized report facility of the software package
Delivery	Output	– Output to file – Output to printer, plotter etc. – Output to other software through interface

With respect to data extraction and aggregation of results, the classification of the literature (in Table 2) should be quite robust because the categories are straightforward both to define and to use. However, summaries of other aspects of the literature were qualitative in nature, based on our own interpretation of the papers, so it is possible that other researchers might come to different conclusions.

4. Conclusions

This study provides an overview of the literature associated with evaluation and selection of software packages. The classification of the literature should be of value both to other researchers who are interested in the state of the art of software package evaluation and selection, and to practitioners who need information about how to evaluate specific types of software package.

Although the methodologies for software selection proposed in different studies usually follow the same process, not much work has been done on establishing a generic methodology which can be used for selection of software package of any type. On the basis of literature review we propose generic software selection methodology and evaluation criteria.

AHP technique has been widely used for evaluation of the software products. Other techniques discussed in the literature to evaluate software products are feature analysis, weighted average sum, expert system, and fuzzy based approach. The most important activities involved in evaluating software packages are: identifying criteria to be considered for evaluation, assigning weights to each criterion, setting up a rating scale for each criterion, calculating the score, and finally to rank the alternatives and select the best one.

Many papers provide a preferred list of evaluation criteria for evaluation of specific software package; however, a lack of common list is apparent. Software evaluation criteria are not clearly defined and elaborated in the literature. The exact meaning of a criterion is open to the evaluator's own interpretation. Sometimes the terminology used by author(s) for a criterion in one literature is different than another author(s) for the same criterion. This may lead to ambiguity and gives unclear picture to the evaluator. To address this issue we provide generic lists of evaluation criteria and their meaning, which can be used for evaluation of any type of the software package. Although, functional criteria for software selection vary from software to software, criteria related to cost, vendor and quality of the software may be common and can be used for selection of any software package. The standard list of evaluation criteria and their definition could overcome some of the pitfalls in software evaluation and selection process.

There is need to develop a framework including a software selection methodology, an evaluation technique, an evaluation criteria, and a system to support evaluation and selection of any software package. As for future work we intend to develop a generic framework comprising a methodology for selecting software package, evaluation criteria, evaluation technique, and knowledge based system/tool which will assist decision makers in evaluating and selecting software packages. For developing a knowledge based system for software evaluation we intend to use a hybrid intelligent approach that employs rule based and a case based reasoning techniques. Such framework will help decision makers not only in evaluation and selection of the software packages but also reduce the time and effort required for software selection.

Acknowledgements

We would like to thank the Editor and anonymous referees of the IST journal for their useful comments to enhance quality of the paper.

References

[1] A. Adhikari, M.I. Lebow, H. Zhang, Firm characteristics and selection of international accounting software, Journal of International Accounting, Auditing and Taxation 13 (2004) 53–69.
[2] E.E. Anderson, Y. Chen, Microcomputer software evaluation: an econometric model, Decision Support Systems 19 (1997) 75–92.
[3] D. Arditi, S. Singh, Selection criteria for commercially available software in constructing accounting, Project Management 9 (1) (1991) 39–44.
[4] S. Bandini, F. Paoli, S. Manzoni, P. Mereghetti, A support system to COTS-based software development for business services, SEKE'02 ACM (2001) 307–314.
[5] J. Bhuta, B. Boehm, A Method for Compatible COTS Component Selection, Springer-Verlag, ICCBSS 2005, LNCS 3412, pp. 132–143.
[6] L.A. Blanc, W.M. Korn, A structured approach to the evaluation and selection of CASE tools, ACM (1992).
[7] L. Blanc, M. Jelassi, DSS software selection: a multiple criteria decision methodology, Information and Management 17 (1989) 49–65.
[8] C. Bozdag, C. Kahraman, D. Ruan, Fuzzy group decision making for selection among computer-integrated manufacturing systems, Computers in Industry 51 (2003) 3–29.
[9] P.Y.K. Chau, Factors used in the selection of packaged software in small businesses: views of owners and managers, Information and Management 29 (1995) 71–78.

[10] J.K. Cochran, H. Chen, Fuzzy multi-criteria selection of object-oriented simulation software for production system analysis, Computers and Operations Research 32 (2005) 153–168.
[11] K. Collier, B. Carey, D. Sautter, C. Marjanierni, A methodology for evaluating and selecting data mining software, in: Proceedings of the 32nd Hawaii International Conference on System Sciences, 1999, pp. 1–11.
[12] E. Colombo, C. Francalanci, Selecting CRM packages based on architectural, functional, and cost requirements: empirical validation of a hierarchical ranking model, Requirements Engineering 9 (2004) 186–203.
[13] S. Comella-Dorda, J.C. Dean, E. Morris, P. Oberndorf, A Process for COTS Software Product Evaluation, Springer-Verlag, ICCBSS 2002, LNCS 2255, pp. 86–96.
[14] L. Davis, G. Williams, Evaluation and selecting simulation software using the analytic hierarchy process, Integrated Manufacturing Systems 5 (1) (1994) 23–32.
[15] X. Franch, J.P. Carvallo, Using quality models in software package selection, IEEE Software 20 (1) (2003) 34–41.
[16] X. Franch, J.P. Carvallo, A quality model based approach for describing and evaluating software packages, in: Proceedings of the IEEE Joint International Conference on Requirement Engineering (RE'02), 2002, pp. 1–8.
[17] G. Grau, J. Pablo Carvallo, X. Franch, C. Quer, DesCOTS: a software system for selecting COTS components, in: Proceedings of the 30th EUROMICRO Conference, IEEE, 2004.
[18] B. Hecht, Choose the right ERP software, Datamation 43 (3) (1997) 56–58.
[19] V. Hlupic, R.J. Paul, Methodological approach to manufacturing simulation software selection, Computer-Integrated Manufacturing Systems 9 (1) (1996) 49–55.
[20] V. Hlupic, A.S. Mann, SimSelect: a system for simulation software selection, in: Proceedings of the 1995 Winter Simulation Conference, 22, 1995, pp. 720–727.
[21] X.B. Illa, X. Franch, J.A. Pastor, Formalising ERP selection criteria, in: Proceedings of the Tenth International Workshop on Software Specifications and Design, IEEE, 2000.
[22] M. Jarkee, Y. Vassiliou, A framework for choosing database query language, Computing Surveys 17 (3) (1985) 313–340.
[23] R. Kathuria, M. Anandarajan, M. Igbaria, Selecting IT applications in manufacturing: a KBS approach, OMEGA 27 (1999) 605–616.
[24] J. Kim, J.Y. Moon, An AHP and survey for selecting workflow management systems, International Journal of Intelligent Systems in Accounting, Finance, and Management 6 (1997) 141–161.
[25] C.S. Kim, Y. Yoon, Selection of good expert system shell for instructional purposes in business, Information and Management 23 (5) (1992) 249–262.
[26] J.D. Kiper, E. Howard, C. Ames, Criteria of evaluation of visual programming languages, Journal of Visual Languages and Computing 8 (1997) 175–192.
[27] B.A. Kitchenham, S. Linkman, D. Law, DESMET: a methodology for evaluating software engineering methods and tools, Computing and Control Engineering Journal (1997) 120–126.
[28] B.A. Kitchenham, Evaluating software engineering methods and tools. Parts 5 to 8, ACM SIGSOFT notes, starting January 1996.
[29] B. Kizzort, Selection of components for OTS component based systems, IEEE (2001).
[30] J. Kontio, G. Caldiera, V.R. Basili, Defining factors, goals and criteria for reusable component evaluation, in: CASCON '96 Conference, Toronto, Canada, November 12–14, 1996.
[31] J. Kontio, A case study in applying systematic method for COTS selection, Proceedings of ICSE-18, IEEE (1996) 201–209.
[32] D. Kunda, STACE: Social Technical Approach to COTS Software Evaluation Component Based Software Quality, vol. 2693, Springer-Verlag, LNCS, 2003. pp. 64–84.
[33] V.S. Lai, R.P. Trueblood, B.K. Wong, Software selection: a case study of the application of the analytical hierarchical process to the selection of a multimedia authoring system, Information and Management 36 (1999) 221–232.
[34] P.K. Lawlis, K.E. Mark, D.A. Thomas, T. Courtheyn, A formal process for evaluating COTS software products, IEEE Proceedings Software 144 (3) (1997) 149–192.
[35] H. Lee, P. Shen, W. Chih, A fuzzy multiple criteria decision making model for software selection, in: IEEE International Conference on Fuzzy Systems, 2004, pp. 1709–1713.
[36] K.R.P.H. Leung, Hareton K.N. Leung, On the efficiency of domain-based COTS product selection method, Information and Software Technology 44 (2002) 703–715.
[37] H.-Y. Lin, P.-Y. Hsu, G.-J. Sheen, A fuzzy-based decision making procedure for data warehouse system selection, Expert Systems with Applications (2006).
[38] S.K. Misra, Analysing CASE system characteristics: evaluative framework, Information and Software Technology 32 (6) (1990) 415–422.
[39] A. Mohamed, T. Wanyama, G. Ruhe, A. Eberlein, B. Far, COTS evaluation supported by knowledge bases, LSO 2004, LNCS 3096, Springer-Verlag, pp. 43–54.
[40] R.P. Mohanty, S. Venkataraman, Use of analytic hierarchy process for selecting automated manufacturing systems, International Journal of Operations and Production Management 13 (8) (1993) 45–57.
[41] M. Mollaghasemi, J. Pet-Edwards, Technical Briefing: Making Multiple Objective Decisions, IEEE Computer Society Press, Los Alamitos, California, 1997.
[42] D. Morera, COTS Evaluation Using Desmet Methodology and Analytic Hierarchy Process (AHP), PROFES 2002, LNCS 2559, Springer-Verlag, pp. 485–493.
[43] M. Morisio, A. Tsoukias, IusWare: a methodology for the evaluation and selection of software products, IEEE Proceedings Software Engineering 144 (3) (1997).
[44] J. Nikoukaran, R.J. Paul, Software selection for simulation in manufacturing: a review, Simulation Practice and Theory 7 (1999) 1–14.
[45] J. Nikoukaran, V. Hlupic, R.J. Paul, A hierarchical framework for evaluating simulation software, Simulation Practice and Theory 7 (3) (1999) 219–231.
[46] J. Nikoukaran, V. Hlupic, R.J. Paul, Criteria for simulation software evaluation, in: Proceedings of the 1998 Winter Simulation Conference, 1998.
[47] E.W.T. Ngai, E.W.C. Chan, Evaluation of knowledge management tools using AHP, Expert System with Applications (2005) 1–11.
[48] K. Oh, N. Lee, S. Rhew, A selection process of COTS components based on the quality of the software in a special attention to internet, HIS 2003, LNCS 2713, 2003, pp. 626–631.
[49] W. Ossadnik, O. Lange, AHP-based evaluation of AHP-Software, European Journal of Operational Research 118 (1999) 578–588.
[50] N. Patel, V. Hlupic, A methodology for the selection of knowledge management (KM) tools, in: Proceedings of the 24th International Conference on IT Interfaces 2002, Cavtat, Croatia.
[51] M. Perez, T. Rojas, Evaluation of workflow type software products: a case study, Information and Software Technology 42 (2000) 489–503.
[52] G.E. Phillips-Wren, E.D. Hahn, G.A. Forgionne, A multiple criteria framework for evaluation of decision support systems, OMEGA 32 (4) (2004) 323.33.
[53] A.L. Plessis, A method for case tool evaluation, Information and Management 25 (1993) 93–102.
[54] C. Reed, A minimum set of criteria for selecting a turn-key geographic information system, Computer and Environment Urban System 7 (1982) 329–333.
[55] B.C. Reimann, A.D. Waren, User-oriented criteria for the selection of DSS software, Communications of the ACM 28 (2) (1985).
[56] G. Rincon, M. Alvarez, M. Perez, S. Hernandez, A discrete event simulation and continuous software evaluation on a systematic quality model: an oil industry case, Information and Management 42 (2005) 1051–1066.
[57] T.L. Saaty, The Analytic Hierarchy Process, McGraw-Hill, New York, 1980.
[58] G.L. Sanders, P. Ghandforoush, A model for the evaluation of computer software packages, Computer and Industrial Engineering 7 (4) (1983) 309–315.
[59] C.-D. Santiago, C. John, M. Edwin, O. Patricia, A process for COTS software product evaluation, ICCBSS 2002, LNCS 2255, 2002, pp. 86–96.
[60] J. Sarkis, S. Talluri, Evaluating and selecting e-commerce software and communication systems for supply chain, European Journal of Operational Research 159 (2004) 318–329.
[61] A. Shtub, I. Spiegler, A. Kapaliuk, Using DSS methods in selecting operations management software, Computer-integrated Manufacturing Systems 1 (4) (1998).
[62] I. Stamelos, A. Tsoukias, Software evaluation problem situations, European Journal of Operational Research 145 (2003) 273–286.
[63] I. Stamelos, I. Vlahavas, I. Refanidis, A. Tsoukias, Knowledge based evaluation of software systems: a case study, Information and Software Technology 42 (2000) 333–345.
[64] A.C. Stylianou, G. Madey, R.D. Smith, Selection criteria for expert system shells: a socio-technical framework, Communications of the ACM 55 (10) (1999).
[65] A. Teltumbde, A framework for evaluating ERP projects, International Journal of Production Research 38 (17) (2000) 4507–4520.
[66] T.W. Tewoldeberhan, A. Verbraeck, E. Valentin, G. Bardonnet, An evaluation and selection methodology for discrete-event simulation software, in: Proceedings of the 2002 Winter Simulation Conference, 2002, pp. 67–75.
[67] M. Toshtzar, Multi-criteria decision making approach to computer software evaluation: application of the analytical hierarchical process, Mathematics and Computer Modeling 11 (1988) 276–281.
[68] E. Triantaphyllou, Multi-Criteria Decision making Methods: A Comparative Study, Springer, 2000.
[69] J. Verville, A. Hallingten, An investigation of the decision process for selecting an ERP software: the case of ESC, Management Decision 40 (3) (2002) 206–216.
[70] I. Vlahavas, I. Stamelos, I. Refanidis, A. Tsoukias, ESSE: an expert system for software evaluation, Knowledge-based Systems 12 (1999) 183–197.
[71] C. Wei, C. Chien, M.J. Wang, An AHP based approach to ERP system selection, International Journal of Production Economics 96 (1) (2005) 47–62.
[72] D. Welzel, H. Hausen, A five step method for metric-based software evaluation-effective software metrication with respect to quality standard, Microprocessing and Microprogramming 39 (1993) 273–276.
[73] J. Wit, W. Herroelen, An evaluation of microcomputer-based software packages for project management, European Journal of Operational Research 49 (1990) 102–139.
[74] P. Wright, Choosing computer based instructional support system: an evaluation/selection model, Computers Education 14 (3) (1990) 217–225.
[75] K. Yoon, C. Hwang, Multiple Attribute Decision-Making: An Introduction, Sage Publisher, 1995.
[76] F. Zahedi, Database management system evaluation and selection decisions, Decision Sciences 16 (1) (1998) 91–116.
[77] F. Zahedi, A method for quantitative evaluation of expert systems, European Journal of Operational Research 48 (1) (1990) 136–147.