

Enzo Colombo · Chiara Francalanci

Selecting CRM packages based on architectural, functional, and cost requirements: Empirical validation of a hierarchical ranking model

Received: 15 February 2003 / Accepted: 3 November 2003 / Published online: 15 January 2004
© Springer-Verlag London Limited 2004

Abstract This study describes a hierarchical ranking model to help the selection of CRM (customer relationship management) packages based on their functional and technical quality. The model is tested empirically by applying the hierarchical analytical process (AHP) to a sample of 42 CRM packages. Results indicate how functionally similar packages can differ substantially in their technical quality and, thus, in their ability to be integrated within a company's information system. The hierarchical model is verified to be dependable, since the quality-based ranking of packages is found to have a low rank-reversal probability as a consequence of managers' uncertainty in weighing the relevance of different quality variables. From a practical standpoint, these results confirm that CRM packages differentiate in measurable quality variables, which can be used by practitioners as a framework to gather and evaluate software-selection information during feasibility analyses.

Keywords Analytical hierarchy process (AHP) · Customer relationship management (CRM) · Software cost · Software selection

1 Introduction

CRM (customer relationship management) suppliers range from top-tier companies, typically targeting large enterprises, to a number of smaller vendors, which offer CRM functionalities as part of a general-purpose IT platform. It is also expected that top-tier companies will

address other market segments when the demand from large enterprises starts decreasing [1]. Consequently, smaller companies will be faced by the alternative between more expensive top-tier CRM packages and cheaper mid-market solutions. Similar to what has happened in the past with ERP (enterprise resource planning) packages, cost differences are likely to encourage an accurate evaluation of corresponding benefits [2]. According to market growth estimates, numerous companies will soon be challenged by this choice [3, 4, 5].

The objective of this paper is to propose and empirically validate a model supporting the selection of CRM packages during feasibility analyses. Traditional COTS (commercial off-the shelf) selection methodologies emphasize functional quality and costs as the primary decision point of view that guarantees the correspondence between software and organizational requirements [6, 7]. This model is novel as it includes technical selection variables measuring the architectural quality of CRM packages. Technical and functional quality variables are provided operating definitions to be measured quantitatively for different CRM packages. Measures are used to rank packages by their overall quality, according to the contextual priorities of decision makers. From a practical perspective, this contributes to feasibility analyses by providing a way to aggregate separate evaluations of individual selection variables into an overall quality assessment and operate a *pre-selection* of CRM packages. In this way, managers can focus more in-depth qualitative analyses on a restricted set of decision alternatives limited to higher packages in the quality ranking.

In the remainder of the paper, the literature on software selection is considered first. Next, the selection model is presented and the hierarchical ranking process that extracts CRM packages is illustrated based on selection variables and decision makers' priorities. Results of empirical testing are reported in Sects. 4 and 5, and, finally, management and research implications of the findings are discussed in Sect. 6.

E. Colombo (✉) · C. Francalanci
Dipartimento di Elettronica e Informazione,
Politecnico di Milano, Piazza Leonardo Da Vinci,
32, 20133 Milano, Italy
E-mail: colombo@elet.polimi.it
Tel.: +39-2-23993457
Fax: +39-2-23993411

2 State of the art

The software selection process can be conceptualized as a sequence of steps through which companies make decisions on the package to be implemented and on their implementation partners [8, 9]. There is substantial agreement among previous research contributions on three methodological steps that compose the software selection process from the initial adoption intent to the final decision and the subsequent start of implementation activities (Fig. 1):

1. *Pre-selection*, aimed at reducing the number of software alternatives to be considered [10, 11, 12]
2. *Analysis*, aimed at obtaining an in-depth understanding of the technological and functional characteristics of pre-selected packages to evaluate their contextual suitability [11, 12, 13]
3. *Negotiation*, aimed at gathering information on the ability of prospective partners to provide pre- and post-implementation support [12, 14, 15]

A decision not to implement CRM can be made at any step, as additional information is gathered, which may change the initial disposition towards implementation. Furthermore, as the selection process develops, the understanding of a company's software requirements can grow and selection steps may need to be iterated accordingly (Fig. 1).

Although selection steps are general, evaluation criteria, variables, and activities vary with the type of package that is considered for implementation. Research on CRM provides a number of case studies exemplifying implementation issues and activities with an exploratory methodological approach [16, 17]. The generalization of these implementation experiences into a CRM selection methodology is an important unanswered research question. Addressing this methodological gap, this paper focusses on pre-selection, as the first software selection step whose output represents the basis for subsequent evaluation activities.

Pre-selection analyses are usually characterized by a methodological orientation towards quantitative over

qualitative evaluation criteria. Qualitative criteria are preferred when analyses are contextual (that is, they refer to a specific application case) and when decision variables are strongly inter-related [11]. The use of quantitative criteria for pre-selection is encouraged instead by the low level of detail that is usually required and is also generally recommended to reduce decision time. Quantitative evaluations of different decision variables can be more easily aggregated into an overall assessment and, in turn, lead to a faster pre-selection decision [8, 9, 18]. Furthermore, a lesson learned from Maiden and Ncube's work on COTS selection is that measurability of selection criteria significantly increases the effectiveness of the selection process [19].

Maiden and Ncube's prevalent quantitative methodological approach makes the majority of pre-selection methodologies output rather than process oriented. Process-oriented methodologies focus on categorizing, sorting, and synchronizing decision-making activities, whose individual execution is mostly supported through qualitative working guidelines [20, 21]. Conversely, output-oriented methodologies are concerned with obtaining a result and supporting evaluation activities in order to make sure that a decision is ultimately made [22]. Pre-selection methodologies typically consider the support to evaluation activities a primary issue that constitutes the foundation to obtain a final decision. Pre-selection methodologies that have been proposed for ERP packages [23], statistical tools [24], and executive decision support systems are all centred around a quantitative appraisal of packages against pre-defined decision criteria.

It should be noted that in pre-selection methodologies decision criteria are specific to the class of packages that is considered for implementation and need to be reconsidered as a new type of software is introduced. On the contrary, the pre-selection process is commonly based on the appraisal and ranking of packages aimed at prioritizing in-depth analyses. A ranking, as opposed to the extraction of a subset of alternatives, allows the analysis phase to take into consideration any number of packages until sufficient information is gathered to make a decision. Methodological support is usually provided to fine tune the ranking of alternatives by varying the

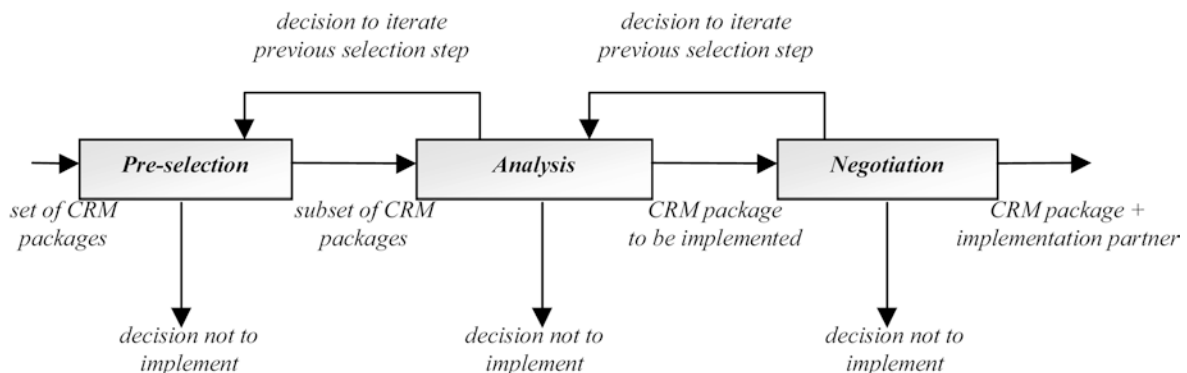


Fig. 1 Software selection steps

importance attributed to different decision criteria or by modifying the selected set of criteria [6, 25].

3 Selection model

Typical decision criteria for pre-selection are costs and quality [20, 26]. In pre-selection, costs are used to fulfil two complementary objectives:

1. To calculate the quality-to-cost ratio and support opportunity evaluations
2. To set a maximum budget for implementation and limit pre-selection to packages satisfying budget constraints

Both objectives share a common attitude towards costs as an organizational control variable that complements rather than determines the ranking of packages. The first objective is achieved if ranking is based on quality and cost is used as a control variable to compare packages with similar levels of quality. In fulfilling the second objective, cost is still used as a control variable to exclude alternatives exceeding a predefined budget. The ranking of remaining alternatives can still be based on quality and complemented by cost evaluations.

A high cost variance can also be observed among CRM packages, probably related to the size of target companies (Table 6). From this perspective, larger companies inevitably involve greater software complexity and consequently incur higher costs. Conversely, smaller companies may not need the most expensive packages and, in this respect, cost limits constitute a legitimate market orientation that precedes the ranking of alternatives.

The empirical analyses reported in Sect. 4 test the correlation between cost and quality. This test is needed

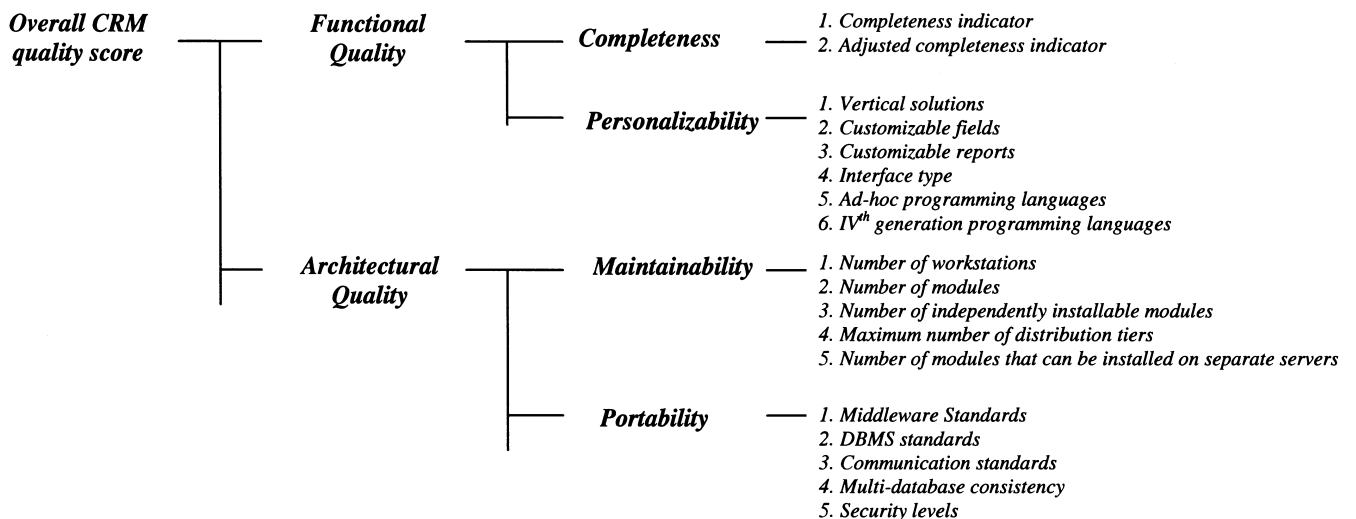
to verify that the inclusion of costs among decision variables does not contribute to ranking and, thus, empirically support the methodological use of costs as a control variable.

Figure 2 describes a model for the evaluation of the quality of CRM packages. The proposed model is novel in that it includes technical software quality variables. Previous software selection research bases the operating measure of quality on the breadth and appropriateness of a package’s functionalities [10], while technical aspects of quality are most often neglected [27]. However, the generally accepted ISO 9126 software quality certification guidelines emphasize both functional and technical quality variables as essential to the effectiveness of applications [24]. Specifically, ISO guidelines define technical quality along three dimensions: *usability*, *maintainability*, and *portability* [28]. The proposed model for pre-selection includes maintainability and portability variables, which are collectively labelled as *architectural* quality decision variables. Usability is not included since it constitutes a more subjective decision criterion that involves a personal assessment of the clarity and learnability of software interaction patterns. This requires knowledge of the specific organizational context which, as observed before, is typically gathered during the subsequent analysis phase for a reduced number of decision alternatives [23].

Similarly, vendor quality variables are not considered because of their strong context dependence. For example, a vendor may be more willing to support larger companies and a general quality assessment of their after-sale assistance could be inaccurate. The expertise and dependability of vendors is also context dependent, since it could largely vary with the particular geographical region and with the capabilities of local implementation partners. For these reasons, vendor-related evaluations are traditionally made through negotiation after an in-depth analysis of pre-selected packages (Fig. 1).

The next sections discuss the determinants of technical and functional quality along dimensions of porta-

Fig. 2 Hierarchical decision model including functional and architectural determinants of quality



bility, maintainability, completeness, and personalizability (Fig. 2). Lower-level determinants of quality represent primary decision variables that can be provided with a quantitative evaluation. These evaluations constitute the basis to obtain an aggregate quantitative measure of high-level decision variables through the hierarchical model discussed in Sect. 4.

3.1 Determinants of architectural quality: portability and maintainability

The traditional exclusion of technical quality variables from pre-selection can be related to multiple concurrent causes. First, technical variables are more difficult to understand for managers, who are typically in charge of the final software selection decision [20]. Furthermore, there exists a widespread management belief that IT variables should not drive decision making, since technical issues can always be solved by purchasing a corresponding technical solution [27].

The IS literature has repeatedly opposed the exclusion of technical variables from management decision-making activities and recognized their impact on software projects' success [27]. The ISO 9126 architectural quality variables included in this paper's pre-selection ranking model (Fig. 2) constitute accepted drivers of project effectiveness [19, 29].

3.1.1 Portability

Packages should be designed in compliance with *modularity* principles [30]. Modularity is technically defined as the ability to independently install and use separate sets of a package's functionalities referred to as *modules*. Different modules interact with each other by exchanging data and services, that is, through their interfaces and without access to each other's private information and procedures. Modularity principles apply to both individual packages and multi-package information systems, where packages act as separate, but interacting complex modules.

Portability is primarily related to the degree of modularity, which is fundamental to integrate packages into a company's information system by parametrizing their interfaces and avoiding cumbersome software re-design activities. Researchers as well as vendors have devoted considerable resources to the definition of standard interface parameters that can guarantee software portability [31]. However, due to the functional complexity of business software, these efforts have resulted in multiple standard interfaces that have been developed in parallel and de facto continue to coexist. Packages can obviate this problem by incorporating multiple standards whose selection is left to designers as a contextual implementation choice. The broader the variety of interface standards supported by a package, the greater the portability.

Common knowledge suggests that standard interfaces are enclosed within *middleware* software. Middleware is defined as a software module that allows heterogeneous systems to interact with each other although individually they may comply with different interface standards. For this purpose, middleware incorporates interoperability paradigms, such as CORBA, DCOM, and RMI (Table 1), and provides translation capabilities from and to different standards. Accordingly, packages are considered more portable if they include a middleware component and if the variety of interface standards supported by their middleware is broad [28].

However, for CRM packages the middleware does not address all relevant aspects of portability. CRM is aimed at providing integrated access to organizational data, which are then exploited to improve a company's relationship with customers. This involves access to distinct and possibly heterogeneous databases managed by different DBMSs (data base management systems). This ability to operate with heterogeneous data standards is an important aspect of portability for CRM packages and the number of different DBMSs that are supported is an additional indication of portability (Table 1).

Note that part of the organizational data may originate from partner companies. Different inter-organizational

Table 1 Summary of portability variables

Name of variable	Definition	Typical values for CRM packages (examples)
Middleware standards	It measures the breadth of middleware standards that are supported by a CRM package.	CORBA, DCOM, RMI, ODBC, JDBC, OLE-DB
DBMS standards	It measures the breadth of data base management systems that can be accessed by a CRM package.	SQL Server, Oracle 8, Informics, DB2, Sybase
Communication standards	It measures the breadth of interorganizational data exchange standards that are supported by a CRM package.	EDI or Edifact, XML
Multi-database consistency	It measures the degree to which replicated and distributed data are aligned and, therefore, consistent over time.	Real-time vs. batch
Security levels	It measures the breadth of security policies that can be supported by a CRM package.	User identification, user profiles, secure communication among data sources (data encryption)

communication standards have developed over time and are still used by organizations, despite the completeness and versatility of more recent data models, such as XML. The ability to support multiple communication standards is also critical to CRM portability and complements the ability to deal with internal data standards (Table 1).

By accessing separate databases, CRM could affect data consistency. Data are typically retrieved from different sources and integrated into a unified schema [31, 32]. Consistency is preserved if modifications to these integrated data are simultaneously translated into updates of the original databases. Delays in updating databases cause temporary misalignments between CRM and original data, which could result in functional errors. For example, a call-centre operator may not be able to view a customer's latest transactions and provide real-time assistance.

This ability to perform real time as opposed to batch data updates depends on the degree of a package's portability. For updates to be real time, CRM modules need to include specific software procedures that align different data sources. If a CRM package does not include these procedures, it cannot be integrated into real-time information systems through predefined software interfaces and would require considerable redesign [33]. In turn, batch as opposed to real-time data consistency is an indication of lower portability (Table 1).

A further aspect of CRM's portability is related to the breadth of security policies that are supported (Table 1). CRM allows access to organizational data through different channels, both internal and external. The ability to support different security policies is critical to the ease of integration of a CRM package within a

company's information system through parametrization as opposed to redesign [33]. For example, a CRM package may not provide built-in user profiles that can be associated with different data access rights. This inability to satisfy user security policies may cause integration difficulties and could represent a cause for low portability.

3.1.2 Maintainability

Modularity is also a fundamental driver of maintainability [28]. In modular packages, changes to a module do not propagate to other modules as long as their interface remains unchanged. Since modules can be independently modified, maintenance is faced with a lower software complexity and can be organized into separate and more manageable tasks [34]. The higher the total number of modules, the lower their individual complexity. Consequently, the total number of modules of a CRM package can be regarded as a first, rough indicator of maintainability (Table 2).

However, modules may not represent independent software entities. The inability to independently install a module stems from an extensive inter-module coupling, which represents a high need for services from other modules [31, 33]. In case of high coupling, changes to a module may have a broad impact on these inter-module service exchanges and the overall maintainability of a package can be lower. The number of modules that can be installed autonomously should be regarded as a further indicator of maintainability (Table 2).

The risk involved with low values of these indicators is to forcibly implement functionalities that are not required by users, but are nonetheless necessary due to

Table 2 Summary of maintainability variables

Name of variable	Definition	Typical values for CRM packages
Number of modules	It indirectly measures the average size of independent code units.	It ranges from 1 for monolithic packages to the maximum number of sub-modules for CRM packages (there is no theoretical limit).
Number of independently installable modules	It measures the level of independence among modules by indicating whether groups of modules or sub-modules need to be simultaneously installed even if only a subset of them is required.	It ranges from 1 for monolithic packages to the maximum number of sub-modules for CRM packages (there is no theoretical limit).
Number of workstations	It measures the maximum number of simultaneous users that can be supported.	The minimum value is typically 5, corresponding to the minimum number of licences that can be purchased. A maximum value may be specified by the vendor for a package (there is no theoretical software limit).
Maximum number of distribution tiers	It measures the ability to split a package into separate application tiers that can be distributed onto different servers.	It ranges from two distribution tiers (traditional client-server architecture) to four (multi-channel architecture with two additional tiers for each channel's specific presentation and application logic, respectively).
Number of modules that can be installed on separate servers (at the same distribution tier)	It measures the ability to distribute modules onto different servers. The level of distribution varies with tiers.	Either supported or not supported by a package's middleware module.

the low modularity and high inter-module coupling of packages. On the contrary, companies can greatly benefit from an incremental deployment of CRM functionalities starting from a reduced set of core modules and subsequently extending support to non-core activities. An incremental deployment also reduces initial investments and allows a more cautious experimentation of CRM technologies. From the authors' experience during this study, to accommodate these requirements, packages with low modularity and high inter-module coupling provide documentation for a guided installation process which allows the deployment of selected functionalities. However, this installation process involves software design activities, as opposed to simple parametrization.

The maintainability of CRM also has hardware determinants. Over time, the number of users can increase and hardware processing capacity has to be upgraded accordingly. These upgrades are enabled by hardware scalability, that is, the potential of a hardware architecture to incrementally grow in size by accommodating increasing requirements with technology changes of comparable magnitude [35]. The inability to satisfy hardware maintainability requirements may result in the obsolescence of the software package, which may have to be replaced to implement a more flexible hardware architecture [23, 36]. CRM packages can facilitate hardware scalability in three ways:

1. Packages can support a varying number of workstations, that is, simultaneous users. The maximum number of workstations that can be supported represents an indicator of maintainability, since it can favour scalability as a higher number of users requires access to CRM functionalities over time (Table 2).
2. Packages can split vertically into *application tiers*. If a package can be split into a higher number of application tiers, hardware scalability is increased, since greater capacity requirements can be satisfied by adding new tiers and corresponding machines (Table 2).
3. Packages can split horizontally, that is, they can allow application modules to be allocated onto separate machines. The number of modules that can be installed on separate machines is an indication of hardware scalability, since increasing capacity requirements can be satisfied by moving modules onto new hardware components (Table 2).

3.2 Determinants of functional quality: completeness and personalizability

Functional quality is traditionally defined as the degree to which software satisfies functional requirements [13]. Packages offer a pre-defined set of functionalities, grouped into modules and sub-modules. Functional quality depends on the *completeness* of a package's

functionalities and on their *personalizability* according to organizational requirements [31].

Functional completeness is emphasized as the main quality variable in the official documentation of CRM packages. Functionalities are usually described by listing and discussing the package's hierarchy of modules and sub-modules. In order to compare the functional structure of different packages, a general classification of CRM functionalities is required. The hierarchical classification of CRM functionalities into modules and sub-modules that has been used in this study is reported in Appendix 1.

There is substantial agreement in the literature about three broad CRM functional areas, *collaborative*, *analytical*, and *operational*, which constitute the foundation of the classification reported in Appendix 1 [31]. Collaborative CRM functionalities allow customers to efficiently and consistently interact with an organization through multiple channels. Analytical CRM functionalities integrate, store, and manage customer information collected through multiple channels to be used by operational CRM functionalities. Operational CRM functionalities support an organization's operations by exploiting CRM data to support planning, marketing, and sale activities. During this study, we have verified that the sub-modules within these three general functional areas constitute the most detailed information that can be obtained from official CRM documentation. Further knowledge should be gathered through demos and interviews and, hence, seems more appropriate for elicitation during in-depth analyses for a subset of alternatives.

Static and dynamic personalizability are distinguished for CRM packages [33]. Some of the personalization requirements can be foreseen by vendors and are accommodated by means of software parameters. However, software parameters can only be assigned a statically pre-defined set of values and personalization requirements that have not been foreseen must be dynamically accommodated by redesigning software code.

Static personalization is typically supported at an aggregate industry level by providing different versions of the package, referred to as vertical solutions, that encapsulate processes and best practices typical of a specific market segment (Table 3). A higher number of vertical solutions is an indication of greater personalizability. At a less aggregate level, interfaces, i.e. paper reports and screen fields, can also be parametrized. Table 3 defines the customizable reports and customizable fields variables accordingly. An interface type variable has also been included in Table 3 to discriminate packages that still fail to include a graphical user interface.

Dynamic personalization involves reprogramming. In this respect, packages may not release their source code to their implementation partners and, therefore, do not allow any form of dynamic personalization. In higher-quality packages, reprogramming is instead allowed by means of ad hoc languages, which are designed to take

Table 3 Summary of personalizability variables

Name of variable	Definition	Typical values for CRM packages
Vertical solutions	It measures the number of customized versions of a package accommodating the typical requirements of a specific industry (that is, embedding an industry's best-practice processes).	It ranges from 1 for general-purpose packages to the maximum number of industries that are addressed (there is no theoretical limit).
Customizable fields	It measures the ability to personalize the layout of a package's interface (that is, to make decisions on the fields to be included and their position on the screen).	Not supported, supported but limited to the system's administrator, supported at an individual user's level.
Customizable reports	It measures the ability to personalize the layout of the reports produced by a package (that is, to make decisions on the items to be included and their position inside a report).	Either supported or not supported by a package.
Interface type	It indicates whether a package includes a graphical interface.	Text based, web based, or windows based.
Ad-hoc programming languages	It measures the ability to personalize modules by means of ad-hoc, high-level programming languages.	Either supported or not supported by a package.
IVth generation programming languages	It measures the ability to personalize modules by means of a standard IVth generation programming language and it indirectly measures the language independence of a package.	It ranges from 0, if suppliers do not release their package's source code to their implementation partners, to the maximum number of programming languages that are supported (there is no theoretical limit).

advantage of a package's modular structure and provide useful high-level instructions (Table 3). Reprogramming can also be supported by standard fourth generation languages. The number of fourth generation languages that are supported is also an indication of quality, as it reduces the risk of skill shortage for a specific language and increases the ability to expand a package's functionalities by integrating code developed in any language (Table 3).

4 Methodology and results

Because of the numerous decision variables to be accounted for, the pre-selection of CRM packages represents a multi-criteria decision making (MCDM) problem [37, 38]. The previous section analysed the problem space and built a theoretical model including two fundamental *points of view* [37], functional quality and technical quality, which, in their turn, are defined by several interconnected *primary evaluation variables*. The second methodological step of MCDM problems is referred to as *evaluation* and is aimed at the empirical testing of the decision model. The objectives of testing MCDM models are:

1. To verify whether primary evaluation variables discriminate decision alternatives
2. To understand how quantitative measures of primary evaluation variables should be weighed into a final score

Recommendations to decision makers should be gathered from testing as a final methodological step of MCDM problems [25]. Evaluation is sound if decision makers can understand how to apply the decision model and obtain a ranking that is consistent with their decision priorities.

4.1 Data sample and descriptive statistics

The sample is comprised of data on 42 CRM packages, including about one-third of the packages available on the global market [39]. Only 20% of the packages in the sample operate in a single country. Extended ERP, knowledge management, and datawarehouse packages have been excluded from the data collection process because of their focus on a limited set of CRM functionalities. Table 4 reports the sample's average value and standard deviation of functional completeness, measured as described in Appendix 2 and normalized in the 0:1 range. All packages in the sample offer functionalities in at least three modules and seven packages out of 42 address all modules. Descriptive statistics of the sample's architectural quality are reported in Table 5, measured as described in Appendix 2 and normalized in the 0:1 range. Note how architectural quality shows higher values of standard deviation than functional quality, suggesting that technical variables may have greater discriminating capability for pre-selection. Table 6 shows descriptive cost statistics.

Table 4 Descriptive statistics of the sample's functional quality

Functional quality variable	Modules	Average quality	Standard deviation
Completeness	Channel management	0.484	0.284
	Call-centre/help desk	0.365	0.314
	Data integration	0.361	0.290
	Data warehousing and knowledge management	0.449	0.301
	Field	0.169	0.321
	Sales	0.288	0.297
	Marketing	0.476	0.283
	Product management	0.238	0.390
Personalizability	Vertical solutions	0.194	0.299
	Customizable fields	0.380	0.180
	Customizable reports	0.300	0.211
	Interface type	0.300	0.250
	Ad-hoc programming languages	0.222	0.427
	IVth generation programming languages	0.486	0.218

Table 5 Descriptive statistics of the sample's architectural quality

Architectural quality variable	Variables	Average quality	Standard deviation
Portability	Middleware standards	0.523	0.211
	DBMS standards	0.604	0.313
	Communication standards	0.466	0.285
	Multi-database consistency	0.333	0.483
	Security levels	0.316	0.211
Maintainability	Number of modules	0.323	0.238
	Number of independently installable modules	0.571	0.395
	Number of workstations	0.535	0.227
	Maximum number of distribution tiers	0.598	0.200

Table 6 Descriptive statistics of the sample's licence costs (figures represent total licence costs for a given number of workstations)

Number of workstations	Average licence costs (U.S. \$)	Standard deviation of licence costs
5	23.652	16.147
15	60.583	45.883
30	107.328	85.355
50	149.603	111.753
100	233.281	147.100
150	296.101	173.276

Data have been obtained primarily through questionnaires that were submitted to vendors and cooperatively filled through telephone and personal meetings with both marketing and production managers. Data have been checked for consistency and validated with information from Internet sites and official documentation of packages. Although the questionnaire was submitted to 120 CRM vendors, 42 demonstrated willing to take part in the study and provided complete information, with the exception of architectural data on the number of modules that can be installed on separate servers at the same application tier (Table 2). This information was provided by only five vendors and, accordingly, the corresponding maintainability variable

has been excluded by empirical analyses. Overall, the data collection and validation effort was carried out over an eight-month period.

4.2 Statistical methodology and approach

According to Roy's classification of MCDM problems, pre-selection constitutes a *choice problem*, which can be defined as *the selection of a subset of satisfactory solutions as an intermediate step to reach a final decision* [40]. Choice problems can be structured hierarchically as shown in Fig. 2 if the following two tests are satisfied [41, 42]:

1. Lower-level criteria should answer *how* the corresponding higher-level criterion can be satisfied by decision alternatives.
2. Higher-level criteria should explain *why* corresponding lower-level criteria should be satisfied by decision alternatives.

Both conditions are verified by the hierarchical decision model in Fig. 2. The first is satisfied by the top-down discussion of primary evaluation variables reported in Sect. 3, which identifies elementary decision criteria as determinants of architectural and functional quality. Conversely, architectural and functional quality

represent fundamental selection criteria that justify the relevance of lower-level variables.

Hierarchical decision models provide a ranking of alternatives by hierarchically combining quantitative assessments of primary evaluation variables into an overall score. Different approaches can be used to combine quantitative assessments of primary evaluation variables. Value or utility functions can be defined to map the measures of each package's primary evaluation variables into an overall score [38]. This approach applies to decision-making models that can benefit from a strong grounding in economic theory which allows the use of pre-defined value or utility functions. Novel or less-structured decision problems typically require an empirical approach [37, 43].

The empirical a priori variant of the *analytic hierarchical process* (AHP) proposed by Saaty [25] is used in this paper to extract an overall score by means of the hierarchical model. The first step of the AHP is the operationalization of model variables, which should be mapped onto a common quantitative scale that supports the hierarchical aggregation of their empirical measures. This involves two methodological steps [37, 44]:

1. Variables whose values are qualitative should be mapped onto a quantitative scale.
2. Different quantitative scales should be normalized onto the same range.

Step one is performed separately for single-valued and multi-valued qualitative variables. The first are mapped onto a quantitative scale by means of pair comparisons among their qualitative values. The output of pair comparisons is an eigenvector of numbers that assesses the impact on CRM quality of different qualitative values. The eigenvector can be used as the corresponding quantitative scale [25, 37]. Multi-valued variables can assume multiple qualitative values simultaneously and a quantitative scale can be obtained if pair comparisons evaluate the impact on quality of all combinations of compatible values, as opposed to individual values only [25]. This approach is feasible if the number of variables to be compared is not too high as in the model shown in Fig. 2 [19]. However, pair-comparison tables do not need to be recalculated when the set of packages to be compared is extended and, thus, guarantee the scalability to a grounding number of selection alternatives. Moreover, a statistical approach is provided to take care of different evaluations related to pair comparisons among selection variables. The analysis is based on the study of the dependability of pair comparisons when multiple actors are involved in the decision process and has the advantage to rank COTS without the use of complex negotiation processes among decision makers as discussed, for instance, by Bohem et al. [15].

Appendix 2 reports the operationalization of model variables, including the pair-comparison tables that have been used to calculate the eigenvectors. A linear normalization function mapping different quantitative scales

onto the 0:1 interval is used in this paper as it is the most common approach in the literature [37]. Note that quantitative variables can have no theoretical limit, while a maximum value should be specified for normalization. In these cases, the maximum empirical value for packages in our sample was selected as the upper bound for normalization. Costs are operationalized as licence costs for different numbers of users ranging from 5 to 150.

Pair comparison tables have been obtained by interviewing 35 senior managers within consulting companies. All interviewees had previous experience with several CRM projects and knowledge of multiple packages. Managers were requested to make comparative evaluations of decision variables by assuming that selection had to be performed for a company operating within the financial industry. They provided different evaluations of decision variables, according to their individual perception of decision priorities. In most cases, they also expressed their individual priorities as a range as opposed to a number, due to both evaluation uncertainty and possible context dependence. This variability in managers' evaluations may translate into changes in the ranking of packages. From a methodological standpoint, if for all values within evaluation ranges there are no rank reversals, ranking can be considered dependable [25]. If, on the contrary, rank reversals occur as evaluations are set closer to the limits of their variability ranges, managers should be provided methodological support by estimating the probability of rank reversals, to be used as an assessment of ranking dependability.

An aggregate comparison matrix was built from managers' individual evaluations by taking their lowest and highest score for each pair of decision variables as the limits of an overall evaluation range. This aggregation process maximizes variability and allows the analysis of ranking dependability within the worst case scenario.

The literature provides empirical evidence showing that by randomly taking values within evaluation ranges, the corresponding elements of the eigenvector have a normal distribution [45]. It has been verified by means of the Kolmogorov-Smirnov test that managers' evaluations comply with these empirical indications by providing weights ω_i with a normal distribution around a mean value μ_i . Rank reversal is tested within a 99% confidence interval around the mean value calculated as $\mu_i \pm 2.59\sigma_i$, according to the standard deviation σ_i of each weight ω_i . Results are reported in Appendix 3. Note that the eigenvectors associated with fundamental points of view, that is, functional and architectural quality, can be analytically determined from the eigenvectors of lower-level variables, as described in Saaty [25].

4.3 Evaluation

The first objective of evaluation is to verify whether decision variables are mutually preferential independent

Table 7 Correlation coefficients among principal components of architectural and functional quality

		Principal components of functional quality		
		1	2	3
Principal components of architectural quality	1	0.389	0.114	0.262
	2	–	0.173	0.173
	3	–	–	–0.415

Table 8 Correlation coefficients among principal components of architectural quality and costs

Principal component	Number of installed stations					
	5	15	30	50	100	150
1	0.836	0.861	0.870	0.868	0.833	0.842
2	–0.172	–0.232	–0.281	–0.261	–0.180	–0.195
3	0.047	0.045	0.057	0.054	0.051	0.051

Table 9 Correlation coefficients among principal components of functional quality and costs

Principal component	Number of installed stations					
	5	15	30	50	100	150
1	0.193	0.224	0.285	0.320	0.318	0.303
2	–0.084	0.061	0.801	0.874	0.925	0.951
3	0.046	0.063	0.125	0.159	0.162	0.147

and, therefore, contribute to the decision process. A *principal component analysis* was conducted separately on primary evaluation variables and fundamental points of view [46, 47]. Table 7 reports the correlation coefficients among the principal components of functional and architectural quality. The correlation coefficients between costs and the principal components of functional and architectural quality are reported in Tables 8 and 9, respectively.

The second evaluation objective is to test the dependability of ranking. An initial ranking is extracted by applying the mean value of weights to the hierarchical model shown Fig. 2. Sensitivity analysis has been conducted at all levels of the hierarchy by determining for each element of the eigenvectors (a) whether there are rank reversals within the 99% confidence interval, (b) the confidence interval that guarantees the absence of rank reversals and (c) the maximum confidence interval where at most one rank reversal occurs. It was verified through sensitivity analyses that allowing one rank reversal provides confidence levels above 90% for all primary evaluation variables, which can be considered dependable. Sensitivity analysis was conducted on the entire sample to address the worst-case scenario. In a real-case scenario, cost requirements can improve ranking dependability by discarding solutions which generate rank reversals.

Note that fundamental points of view do not require sensitivity analyses, as architectural and functional quality have been evenly weighed by all managers. Portability, maintainability, completeness, and personalizability, that is, middle-level criteria in the hierarchical model, were weighed differently by managers. However, managers provided only four distinct configurations of decision priorities. Sensitivity has been tested separately for these four configurations, as opposed to analysing a single evaluation range, as is recommended in the literature for higher-level decision criteria.

As a consequence of a rank reversal, a package moves upwards or downwards by one or multiple ranking positions. This lower dependability caused by rank reversals can be addressed by considering a greater number of packages for in-depth analyses, evaluating to the number of ranking positions either acquired or lost by a package plus one. Table 10 reports the overall number of packages that should be considered for in-depth analyses.

5 Discussion of findings

A first result concerns the contribution of both functional and architectural decision variables in determin-

Table 10 Minimum number of alternatives to be selected for in-depth analysis

Managers' priorities	Second level of the hierarchy			Overall model
	Number of reversals	Number of ranking positions changed for each reversal	Number of solutions to select	Number of solutions to select ^a
Pers. = Compl Portab. = 2 Maint.	2	{1,1}	Max({1,1}) + 1 = 2	Max({1,2}) + 1 = 3
Pers. = 2 Compl Portab. = 2 Maint.	5	{1,1,2,1,1}	Max({1,1,2,1,1}) + 1 = 3	Max({1,2,2,1,1}) + 1 = 3
Pers. = Compl 2 Portab. = Maint.	1	{3}	Max({3}) + 1 = 4	Max({3}) + 1 = 4
Pers. = Compl 2 Portab. = Maint.	2	{1,2}	Max({1,2}) + 1 = 2	Max({1,2}) + 1 = 3

^a Adjusted according to reversals due to variability at the lowest level of the hierarchy (overall confidence evaluates to 90%)

ing the overall score of CRM packages. The principal components of functional and architectural quality have been found to be mutually preferential independent, suggesting the inclusion of both points of view in the selection process. Descriptive statistics have also shown that the variance of both aspects of quality is high among CRM packages. As a consequence, high functional quality may be accompanied by low architectural quality, which, in turn, could raise numerous technical obstacles, whose negative impact on implementation and maintenance activities have been extensively discussed in the literature [31, 33]. In particular, it is important to recall that limited portability and maintainability have been found to cause design difficulties whose solution often involves the replacement of the package [23, 36].

5.1 Findings

The results above provide preliminary evidence of the applicability and dependability of the proposed hierarchical selection model. Costs have been found to be positively correlated with architectural quality (Table 8). This supports their use as a control rather than a selection variable and indicates a first practical consequence of a cost-minimization approach to the selection of CRM packages. Lower costs are likely to be accompanied by inferior architectural quality, which, as a source of technical difficulties, may generate unexpected expenses as the project unfolds.

On the other hand, budget limits can represent a necessary constraint. Using costs as a control variable limits pre-selection to packages satisfying budget constraints. Companies can partly release these constraints as they examine pre-selected packages against their high-level architectural requirements, according to an iterative approach to pre-selection. Alternatively, they can accept limitations on selected architectural variables given their specific organizational requirements and extract lower-cost packages for subsequent in-depth analyses.

The principal components of costs and functional quality do not show significant levels of correlation. A possible explanation for this lack of correlation can be related to the low average level of functional completeness of packages, reported in Table 4. Packages tend to focus on a few modules, for which they offer a relatively high number of sub-modules. Other modules show significantly lower levels of functional completeness, although they are seldom totally missing. As a consequence, packages with a low functional completeness can be highly specialized and, thus, have sufficient competitive advantage to obtain high license fees.

However, the ranking of packages is not meant to determine the final selection decision, but represents a support to pre-select a restricted subset of packages to

be analysed in-depth (Sect. 2). Provided that multiple packages are pre-selected for in-depth analysis, one or a few rank reversals can be acceptable. For example, the uncertainty caused by one rank reversal is obviated if a minimum subset of two packages is pre-selected for in-depth analysis. Table 10 shows that at most four packages should be selected for in-depth analyses, depending on the configuration of the comparison matrix that is considered. This number seems reasonable with respect to managerial practices reported in case studies. Furthermore, packages may be eliminated from ranking if they do not fulfil either functional or architectural constraints. As a side effect, this may reduce the number of rank reversals and, consequently, the minimum number of packages to be pre-selected for in-depth analyses. Overall, the model seems sufficiently dependable with respect to empirical levels of uncertainty in weighing decision variables.

5.2 Recommendations to decision makers

Our findings have important implications for CRM feasibility analyses. First of all, decision makers should be aware that marketing information provided by CRM vendors is incomplete if it exclusively focusses on functional characteristics of packages. In this respect, it should be observed that this current study has collected information on the architectural characteristics of packages by means of ad hoc questionnaires, while functional information was available either from vendors' Internet sites or standard marketing documents. Primary decision variables provided by this study can be used as a basis to gather more accurate information to support CRM software selection.

Results encourage a rigorous methodological approach to obtain an overall evaluation of packages' quality based on complete pre-selection information. Table 4 shows how there is a wide variance in how CRM packages cover different functional areas. Similar variations are shown by architectural quality variables in Table 5. The high number of quality variables and their mutual independence makes an overall judgment difficult to reach intuitively. Managers have demonstrated that even their comparative judgment of pairs of variables is subject to uncertainty, although they have been asked to provide their contextual priorities for a specific industry. The hierarchical model can help managers cope with their uncertainty by delivering an overall quality evaluation that is aggregate and dependable notwithstanding judgment difficulties.

The model can be applied according to different approaches. First, managers can simply rank CRM packages by using the set of hierarchical weights estimated in this study, which have been verified to provide a dependable ranking if at least four packages are pre-selected for in-depth analysis (Table 10). However, pair comparison tables and, consequently, sensitivity analy-

ses are referred to the financial sector (Sect. 4.3). Managers operating in a different industry may disagree with the decision priorities reported in this paper. In this case, they should redefine pair comparison tables through interviews, or brainstorming sessions or, simply, their own judgment, according to their perceived level of uncertainty [45]. In turn, this redefinition will involve a new estimate of hierarchical weights and of the model's dependability.

Alternatively, they can apply a hybrid approach, partially re-estimating weights. This approach is based on the observation that uncertainty in managers' priorities decreases at higher levels in the hierarchical model. Therefore, managers may re-estimate lower-level comparison matrices, while reusing higher-level weights, possibly improving the model's dependability. Table 10 reports the minimum number of packages to be pre-selected for different priorities in weighing portability with respect to maintainability, and completeness with respect to personalizability. For most configurations of managers' priorities with middle-level variables, the minimum number of packages decreases from four to three.

Note that the numerical distance among overall quality scores cannot be used as a criterion to further reduce the number of packages to be pre-selected [44]. For example, if the first two packages in ranking are scored 1 and 0.1, respectively, it cannot be concluded that the second package should not be considered for in-depth analysis since its quality is ten times lower. The model provides an evaluation of quality according to an ordinal scale, as a consequence of the inclusion of both quantitative and qualitative decision variables. Therefore, the concept of distance among scores cannot be defined. The only methodological criterion that determines the number of packages to be pre-selected is dependability.

As a last observation, functional completeness is defined as the total number of sub-modules provided by a package according the classification reported in Appendix 1. This indicator can be refined as the model is applied within a specific organizational context by assessing the contingent importance of different functionalities through pair comparisons among modules and sub-modules. Thus, a more reliable, context-dependent indicator of functional completeness can be obtained, which may enhance the model's dependability and, hence, the decision process.

6 Concluding remarks

Results show that CRM packages significantly differ in their functional and architectural quality and a dependable ranking can be obtained to support pre-selection. If pre-selection is exclusively based on functional quality, in-depth analyses would be faced with the comparison of functionally similar packages with a

different architectural quality. It could be argued that a company requires selected instances of architectural variables, depending on legacy software components that need to interact with CRM. Unfortunately, excluding architectural variables from a definition of quality represents an oversimplification for many reasons. Architectural standards may not be mutually exclusive and higher-quality standards are compatible with the majority of legacy components. Irrespective of specific requirements, greater architectural quality should be preferred. Over time, requirements typically evolve and only higher-quality architectural solutions can accommodate change.

The set of architectural variables of the study is grounded in the technical literature and it is rather standard. The contribution of this paper is to take a first step towards translating technical knowledge into aggregate, understandable and measurable criteria for feasibility analyses. Initial design phases are especially critical as they require interdisciplinary knowledge to bridge management and technical areas of expertise towards common evaluations and decisions. In subsequent phases, projects are organized into more specialized tasks where expertise is developed. A challenge for both theory and practice is to mine this specialized know-how and gather high-level knowledge for managerial decision making.

This study focusses on the pre-selection step of feasibility analyses, but a number of questions still remain unanswered, and research in this direction would certainly represent a fruitful area of investigation. The inclusion of technical variables should be extended to subsequent software selection phases. In particular, in-depth analyses need support to evaluate packages against organizational requirements and require context-dependent criteria to assess architectural quality. Technical determinants of the need for process change should also be identified to help feasibility analyses. The model could then be generalized to other software packages and the set of decision variables proposed in this paper could be extended according to other packages' specific characteristics, thus filling the literature gap identified in this paper across different functional areas of software architectures.

Acknowledgements This work has been partially supported by the Italian VISPO and MAIS projects funded by the Italian Ministry of Research (MIUR). Particular thanks are expressed to Paolo Galli and Andrea Duro for their assistance in the data collection effort.

7 Appendix 1. CRM modules and sub-modules

7.1 Collaborative CRM

(Tables 11 and 12)

Table 11 Module: Channel management

Sub-modules	Description
System of response	
Interactive voice response system (IVR)	It automatically responds to customers' calls by interacting through voice or tone recognition.
Automated number identification	It associates the caller's number with a customer code, if possible.
Call routing	It routes calls to the most appropriate operator or to the IVR according to predefined rules.
Out-bound functionalities	It generates contact lists and allocates them to the most appropriate contact centre operator.
In-bound functionalities	It allocates in-bound calls to the appropriate operator by balancing the overall workload of the call centre.
Multimedia services	
E-mail classification	It classifies e-mails by means of keyword-based intelligent functionalities.
E-mail routing	It forwards e-mails to the most appropriate operator according to predefined rules.
Automatic e-mail responder	Keyword-based selection of predefined e-mail responses.
E-mail form generation	It provides a set of predefined e-mail models to facilitate and standardize responses.
Chat services	It allows a real-time text interaction between customers and operators, by supporting data sharing.
Fax management	It receives, sends, and stores faxes in electronic format.
Web-CRM connection services	Middleware services that integrate CRM modules with pre-existing web-based functionalities.
Web call back	It allows customers to request assistance in voice over IP mode.
VOIP (voice over IP)	It allows voice and data integration through the IP protocol.
WAP traffic management	It manages in-bound and out-bound WAP traffic.

Table 12 Module: Call centre/help desk

Sub-modules	Description
System of response	
Computer-telephony integration	Based on the number identification module, it automatically places customer information on the operator's PC according to predefined data extraction and formatting rules.
Customer data management	
Trouble ticketing	It stores historical data on past interactions with customers. Interactions aimed at solving the same customer issue are grouped and associated with an identifying "ticket".
Multimedia services	
Co-browsing	It allows contact centre operators to have a real-time view of customers' actions on the organization's Internet site.
Page pushing	It allows operators to send customers one or more pages from the organization's Internet site.
Data sharing	It allows operators to send documents to customers in multiple formats.
Smart script (decision trees)	It helps contact-centre operators to find the solution to customers' issues by means of decision trees.
Quality management	
Work load forecasting	It forecasts system work load, based on in-bound and out-bound calls.
Staffing and scheduling	It manages the contact centre shifts according to traffic forecasts.
Tracking and real-time adherence	It monitors contact centre activities, by tracking call traffic and operators' performance.
Up sell and cross sell	It supports contact-centre operators in up sell and cross sell initiatives.

7.2 Analytical CRM

(Tables 13 and 14)

Table 13 Module: Data integration

Sub-modules	Description
Real-time customer data availability	It integrates customer data stored in separate databases and provides operations with real-time access capabilities.
Data consistency management	It synchronizes and integrates data from different devices and channels.
Integration with the configuration engine	Middleware functionalities that integrate the pre-existing ERP configuration engine with CRM modules.
Back-office integrator	It allows the integration between the CRM package and pre-existing back-office applications.
Interface with knowledge base system	Group of standard and/or proprietary interfaces to integrate the CRM system with pre-existing specific knowledge bases.
Integration with mobile devices of sales agents	It provides sales agents with access to organizational information through mobile devices.

Table 14 Module: Datawarehousing and knowledge management

Sub-modules	Description
Segmentation analysis of customers and market	It provides predefined queries for typical marketing initiatives that support the generation of campaigns.
Multidimensional analysis	It aggregates organizational data according to significant dimensions, such as classes of customers, product types, geographical areas, etc., to support marketing decisions.
Marketing tools	It provides tools to perform customer analysis and comparisons between actual and estimated market trends.
Data mining	It supports the extraction of useful information from large quantities of organizational data
Customer satisfaction analysis	It monitors interaction with customers and calculates indicators such as: average response time to calls, percentage of automatic e-mails that have not solved customers' problems, percentage of customers that have not found the required information from internet site, etc.
Click stream analysis	It supports the identification of customer profiles by analysing their browsing behaviour and habits.
Knowledge base	It supports the creation of organizational knowledge on solving customer problems.

7.3 Operational CRM

(Tables 15, 16, 17, and 18)

Table 15 Module: Field

Sub-modules	Description
Performance monitoring	It tracks and monitors the performance of field service operators. It also defines balanced workloads for each team of operators.
Spare-part management	It helps enterprises to manage and control their spare-part warehouses.
Intelligent assignment and dispatch	It supports contact-centre operators in routing requests to the most appropriate team of field operators.
Service account	It transfers data from the laptops and palms of field operators to a database that is then used to prepare invoices.
Integration with mobile devices	It supports a bi-directional information flow between contact-centre operators and field operators.

Table 16 Module: Sales

Sub-modules	Description
Productivity tools	
Opportunity management systems (OMS)	It provides information about prices, contacts, products, and competitors to identify and exploit sale opportunities.
Account and contact management	It allows vendors to organize the layout and contents of their account and contact management application interface.
Visual query tools	It allows the collection of customized information by supporting the construction of complex queries.
Sale proposals (*)	It supports the production of sale proposals by providing a set of predefined masks for different types of contacts.
Sale proposals with enclosures (*)	It integrates enclosures such as tables, graphs, pictures, and so on with sale offers.
Pipeline management	It allows the extraction of information from a knowledge base about the different players involved in each sale transaction in order to coordinate multiple cooperating executors.
Workforce management and control	
Workforce management and control	It monitors the behaviour of the sale workforce and verifies their performance against their individual objectives.
Territory management	It optimizes the allocation of the sale workforce to the territory.
Budget and sales forecasts	It allows vendors to associate a success probability with their commercial initiatives and simulate different sale scenarios.
Management of the sale channels	
Transaction notification	It allows the automatic communication of the end of a transaction process.
Up sell/cross sell	It supports contact centre operators in up sell and cross sell initiatives.

Table 17 Module: Marketing

Sub-modules	Description
Telemarketing	
Telemarketing campaigns	It selects customer profiles to be involved in telemarketing initiatives.
Strategic Support	
Competitor information processing	It monitors activities that involve competitors.
Marketing report generation	It produces marketing reports through different channels such as fax, printer, and Internet.
Hyperlinks and details	It produces marketing reports with web-format hyperlinks
Campaign management	It manages commercial campaigns by scheduling tasks, outputs, employees, and customers to be involved at different steps.
Budget control	It monitors marketing activities against economic budget indicators.
E-mail generation and one- to-one marketing	It manages campaigns customized to specific customer profiles.
Incentive plans	It provides functionalities to determine incentive plans aligned with organizational goals.
Return on investment (ROI)	It monitors commercial investments through the real-time calculation of the ROI.

Table 18 Module: Product management

Sub-modules	Description
Contact management	
Product configurator (*)	It allows the configuration of products using functional, architectural, and cost parameters in order to verify feasibility and delivery time.
Catalogue of components, sub-components, available platforms, and prices (*)	It provides a multimedia demo of organizational products and services.
Real-time compatibility control	It supports the <i>Product configurator</i> and <i>Catalogue</i> modules by verifying the technical compatibility of product components.
Product configurator on site	It provides product configuration functionalities on vendors' laptops and palms.
Multi-currency and multi-price functionalities	It provides other CRM modules with functionalities needed to account for local currencies.
On-line direct product configuration	On-line public product configuration.
Contact Management	
Inventory checking	Real-time control of product availability.
Order status and delivery confirmation	It provides real-time information about the status of orders including delivery activities.

8 Appendix 2. Operating definition of model variables

(Tables 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29)

Table 19 Determinants of architectural quality

Quality dimension	Primary selection variable	Domain ^a	Numerical values ^a
Portability	Middleware standards	$[1, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbb{N}^+ \cap [1, \text{Max}_{\text{Benchmark}}]\}$
	DBMS standards	$[1, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbb{R}^+ \cap [1, \text{Max}_{\text{Benchmark}}]\}$
	Multiple-database consistency	$[0,1]$	0: Communication – batch update Integration – Real-time update
Maintainability	Communication standards	$[0,1]$	See Table 26
	Security level	$[0,1]$	See Table 27
	Number of stations	$[5, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbb{N}^+ \cap [5, \text{Max}_{\text{Benchmark}}]\}$
	Number of modules and per-module sub-modules	$[1, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbb{N}^+ \cap [1, \text{Max}_{\text{Benchmark}}]\}$
	Number of independently installable modules and sub-modules	$[1, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbb{N}^+ \cap [1, \text{Max}_{\text{Benchmark}}]\}$
	Maximum number of distribution tiers	$[2, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbb{N}^+ \cap [2, \text{Max}_{\text{Benchmark}}]\}$

^a $\text{Max}_{\text{Benchmark}}$ indicates the maximum empirical value of variables within the study sample

Table 20 Determinants of functional quality

Quality dimension	Selection variable	Domain ^a	Values ^a
Completeness	Completeness indicator	[0,1]	$\in \{\mathbf{R}^+ \cap [0,1]\}$
	Adjusted completeness indicator	[0,1]	$\in \{\mathbf{R}^+ \cap [0,1]\}$
Personalizability	Ad-hoc languages	[0,1]	0: Not supported 1: Supported
	Customizable reports	[0,1]	See Table 28
	Interfaces typology	[0,1]	See Table 29
	Customizable fields	[0,1]	See Table 28
	IV-Generation languages	$[0, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbf{N}^+ \cap [0, \text{Max}_{\text{Benchmark}}]\}$
	Vertical solution	$[0, \text{Max}_{\text{Benchmark}}]$	$\in \{\mathbf{N}^+ \cap [0, \text{Max}_{\text{Benchmark}}]\}$

^a Max_{Benchmark} indicates the maximum empirical value of variables within the study sample

Table 21 Paired comparisons between functional and architectural quality (first level of the hierarchical model)

Overall score	1	2
Functional quality	1	1
Architectural quality	–	1

Table 22 Pair comparisons between a) maintainability and portability according to functional quality and b) completeness and personalizability (second level of the hierarchical model)

a) Functional quality	1	2	b) Architectural quality	1	2
Maintainability	1		Completeness		
Portability	1/2;1;2	1	Personalizability	1/2;1;2	

Table 23 Pair comparisons among personalizability variables (third level of the hierarchical model)

Personalizability	1	2	3	4	5	6
Ad-hoc programming languages	1	2	[2,4]	[2,4]	[2,4]	1/2
IVth generation languages		1	[1,2]	[1,2]	[1,2]	1
Customizable reports			1	1	1	[1/8, 1/4]
Interfaces typology				1	1	[1/8, 1/4]
Customizable fields					1	[1/8, 1/4]
Verticalization						1

Table 24 Pair comparisons among maintainability variables (third level of the hierarchical model)

Maintainability	1	2	3	4	5
Middleware standards	1				
DBMS standards	[1,2]	1			
Multi-DB consistency	[2,4]	[2,4]	1		
Communication standards	[1,2]	[1,2]	.5	1	
Security level	[1,2]	[1,2]	[1,2]	1	1

Table 25 Pair comparisons among portability variables (third level of the hierarchical model)

Portability	1	2	3	4
Number of workstations	1	[2,4]	2	1
Number of modules		1	1/2	[1/4, 1/2]
Number of independently installable modules			1	1/2
Number of distribution tiers				1

Table 26 Pair comparisons among qualitative values of *Communication standards*

Communication standards	1	2	3	4
Lack of support	1			
EDI or EDIFACT	2	1		
XML	8	4	1	
EDI or EDIFACT + XML	16	8	2	1

Table 27 Pair comparisons among qualitative values of *Security levels*

Security levels	1	2	3	4
Lack of security support				1
Access control by password but no user profiles			2	1
Access control using user profiles (each profile is enabled to use only a part of the application)			4	2
User profiles + security mechanism for data exchange	16	8	4	1

Table 28 Pair comparisons among qualitative values of *Customizable fields* and *Customizable reports*

Customizable reports/ Customizable fields	1	2	3	4
Not available				1
Reprogramming required		2	1	
Administrator level		4	2	1
Single user		16	8	4

Table 29 Pair comparisons among qualitative values of *Interface type*

Interface type	1	2	3
Text interface	1		
Web-based interface	8	1	
Window interface	16	2	1

9 Appendix 3. Empirical results of pair comparisons

(Tables 30, 31, and 32)

Table 30 Mean value, standard deviation, and 99% confidence interval of the personalizability eigenvector's weights

Personalizability	Distribution data		99% confidence interval	
	Mean	Std. dev.	Lower bound	Upper bound
IVth generation programming languages	0.231	0.011	0.201	0.260
Ad-hoc programming languages	0.133	0.007	0.116	0.151
Interface type	0.078	0.006	0.062	0.095
Customizable report	0.079	0.006	0.064	0.094
Customizable fields	0.078	0.005	0.065	0.092
Vertical solution	0.340	0.018	0.294	0.385

Table 31 Mean value, standard deviation, and 99% confidence interval of the scalability eigenvector's weights

Maintanability	Distribution data		99% confidence interval	
	Mean	Std. dev.	Lower bound	Upper bound
Number of workstations	0.351	0.013	0.318	0.384
Number of modules	0.114	0.010	0.089	0.139
Number of independently installable modules	0.190	0.003	0.181	0.199
Maximum number of distribution tiers	0.345	0.013	0.312	0.378

Table 32 Mean value, standard deviation, and 99% confidence interval of the interoperability eigenvector's weights

Portability	Distribution data		99% confidence interval	
	Mean	Std. dev.	Lower bound	Upper bound
Middleware standards	0.137	0.011	0.109	0.165
DBMS standards	0.131	0.012	0.101	0.161
Multi-DB consistency	0.239	0.013	0.204	0.273
Communication standards	0.262	0.011	0.234	0.289
Security level	0.232	0.014	0.196	0.268

References

- Cahners in-stat group (2001) CRM revenues worldwide moving on up; Europe and Asia Pac to get their piece of the pie. <http://www.instat.com>
- Desmond M (2001) CRM Software: customer service for a song. *PCworld*. <http://www.pcworld.com>
- IDC (2000) Customer relationship management market forecast and analysis, 2000–2004. Report no. W22401. <http://www.idc.com>
- IDC (2002) IDC says Asia/Pacific CRM solutions market will grow by 30%. <http://www.idc.com.sg/Press/2002/AP-PR-crm.htm>
- King J (2001) Premier 100: CRM still in formative stages for many users. *Computerwords*. <http://www.computerwords.com>
- Karlsson J, Ryan K (1997) A cost–value approach for prioritizing requirements. *IEEE Softw* 14(5):67–74
- Sivzattian S, Nuseibeh B (2001) Calibrating value estimates of requirements. In: Third international workshop on economics-driven software engineering research. Co-located with the international conference on software engineering (ICSE)
- Barua A, Ravindran S, Whinston AB (1997) Efficient selection of suppliers over the Internet. *J Manag Inf Syst* 13(4):117–138
- Tam KY, Hui KL (2001) A choice model for selection of computer model vendors and its empirical estimation. *J Manag Inf Syst* 17(4):97–124
- Nikoukaran J, Paul RJ (1999) Software selection for simulation in manufacturing: a review. *Simulation Practice Theory* 7(1):1–14
- Ochs M, Pfahl D, Chrobok-Diening G, Nothhelfer-Kolb B (2001) A method for efficient measurement-based COTS assessment and selection method description and evaluation results. In: Proceedings of the seventh international software metrics symposium (METRICS 2001), pp 285–296
- Dean J, Oberndorf P, Vidger M, Abts C, Erdogmus H, Maiden N, Looney M, Heneiman G, Guntersdorf M (2001) COTS workshop: confirming collaboration for successful COTS development. *ACM SIGSOFT Softw Eng Notes* 26(1): 61–73
- Choi S J, Scacchi W (2001) Modeling and simulating software acquisition process architectures. *J Syst Softw* 59(3):343–354
- Blecherman B (1999) Adopting automated negotiation. *Technol Soc* 21:167–174
- Boehm B, Grunbacher P, Briggs RO (2001) Developing groupware for requirements negotiation: lesson learned. *IEEE Softw* 18:46–55
- Khirallah K (2001) CRM case study. Optimizing relationships at National Australia Bank, Ltd. TowerGroup
- Compton J (2001) Case study: calling on CRM. *ZdnetIndia*. <http://www.zdnetindia.com>
- Anderson EE (1990) Choice models for the evaluation and selection of software packages. *J Manag Inf Syst* 6(4):123–138
- Maiden NA, Ncube C (1998) Acquiring COTS software selection requirements. *IEEE Softw* 15(2):46–56

20. Ahrens JD, Prywes N, Lock E (1995) Software process reengineering: toward a new generation CASE technology. *Syst Softw* 30:71–84
21. McChesney IR (1995) Towards a classification scheme for software process modeling approaches. *Inf Softw Technol* 37(7):363–374
22. Cash JI, Lawrence PR (1990) The information system research challenge: qualitative research methods. Harvard Business School Press, Cambridge
23. Taudes F, Mild A (2000) Options analysis of software platform decisions: a case study. *MIS Q* 24(2):277–243
24. Ossadnik W, Lange O (1999) AHP-based evaluation of AHP-software. *Eur J Oper Res* 118:578–588
25. Saaty TL (1990) How to make a decision: the analytic hierarchy process. *Eur J Oper Res* 48(1):9–26
26. Farbey B, Finkelstein A (2001) Software acquisition: a business strategy analysis. In: Proceedings of the fifth IEEE international symposium on requirements engineering, pp 76–83
27. Orlikowski WJ, Iacono CS (2001) Research commentary: desperately seeking the “IT” in IT research – a call to theorizing the IT artifact. *Inf Syst Res* 12(2):121–134
28. ISO/EIC 9126 (1991) Information technology – software product evaluation – quality characteristics and guidelines for their use. International standard, ISO, Genève, Switzerland
29. Chechik M, Gannon J (2001) Automatic analysis of consistency between requirements and designs. *IEEE Trans Softw Eng* 27(7):651–672
30. Blume M, Appel AW (1999) Hierarchical modularity. *ACM Trans Program Language Syst* 21(4):813–847
31. Greenberg P (2001) CRM at the speed of light. Osborne/McGraw-Hill
32. Bradshaw D, Armstrong S (2001) Ovum evaluates: eCRM. Ovum.<http://www.ovum.com>
33. Chorafas DN (2001) Integrating ERP, CRM, supply chain management and smart materials. Auerbach
34. Booch G (1986) Object-oriented development. *IEEE Trans Softw Eng* 12(2):211–221
35. Chau P, Tam KY (1997) Factors affecting the adoption of open systems: an exploratory study. *MIS Q* 21(1):1–24
36. Berndt DJ, Hevner AR (1997) The COR model for analyzing information systems change. In: Proceedings of the 30th Hawaii international conference on system sciences, vol 3, pp 198–207
37. Saaty TL (1980) The analytic hierarchy process. McGraw-Hill, New York
38. Keeney RL, Raiffa H (1993) Decisions with multiple objectives. Cambridge University Press
39. Ticehurst J (2000) CRM vendor market faces massive shake-out. In: Gartner group European symposium, Cannes
40. Roy B, Bouyssou D (1993) Aide Multicritère à la décision: Méthodes et Cas. Economica, Paris
41. Gibson J (1994) How to do system analysis. Prentice Hall, Englewood Cliffs, NJ
42. Mollaghasemi M, Pet-Edwards J, Gupta U (1995) A multiple criteria buy versus lease analysis for government contracts. *IEEE Trans Eng Manag* 42(3):278–287
43. Dubois D, Prade H, Testemale C (1988) Weighted fuzzy pattern matching. *Fuzzy Sets Syst* 28(3):313–331
44. Fenton NE (1996) Software metrics, a rigorous approach. Thomson Computer Press
45. Saaty TL, Vargas LG (1987) Uncertainty and rank order in the analytic hierarchy process. *Eur J Oper Res* 32:107–117
46. Jolliffe IT (1986) Principal component analysis. Springer, New York Berlin Heidelberg
47. Everitt BS, Dunn G (2001) Applied multivariate data analysis. Oxford University Press, New York