A New Way to Determine External Quality of ERP Software

Ali Orhan Aydin

Department of Computer Engineering
Gelisim University, Istanbul, Turkey
aliorhanaydin@gmail.com

Abstract. Today many production systems plan to use Enterprise Resource Planning (ERP) software to gain competitive advantage. ERP promises in improving efficiency of business processes. However, inappropriate software selection results in a very complex managerial problem which is implementation of ERP software. To reduce the relevant risk, ERP software purchasers need to determine conformance of the software to their requirements. This study aims to define the requirement levels of external quality characteristics and provide a guide to production systems to evaluate ERP software in a systematic manner. In the frame of this reference, a way of evaluating external quality of ERP software is put forward to reduce the risk taken before purchasing it.

Keywords: External Software Quality, Enterprise Resource Planning.

1 Introduction

Change in the market conditions increase competition among production systems since 1970s [1]. Shorter product lifetimes, high quality requirements, demanding customers and availability of diverse alternatives are significant factors that affect current market conditions. These competitive market conditions caused production systems to seek solutions to survive [2].

By using Information Technologies (IT), companies aim to get competitive advantage. In the beginning, organizations tried to utilize applications of Material Requirement Planning (MRP), Manufacturing Resources Planning (MRP II), Distribution Resources Planning (DRP), Capacity Requirements Planning (CRP) and Computer Integrated Manufacturing (CIM). The main objective is to use their resources more effectively and increase competition strength [3].

At the end of the 1980s organizations seek solutions to integrate these types of information systems to manage their flexible systems, supply customized product demand, control product complexity and plan resources more effectively by the use of IT [4]. Integration is considered as a key factor for getting advantage and Enterprise Resource Planning (ERP) begin to be used in organizations in early 1990s [3, 5, 6].

As stated by Davenport [7] ERP systems promise to restructure business processes of organizations, because they intend to solve problems caused by the lack of coordination between applications and business processes [8]. Moreover, those enterprise systems integrate all of the information that flow organization wide [9].

B. Murgante et al. (Eds.): ICCSA 2012, Part IV, LNCS 7336, pp. 186–201, 2012.

Today, ERP software and ERP software implementation market became nearly 50 billion EURO (€) [8]. Day after day, ERP software market grows and many organizations want to benefit from this software. On the other hand, after spending a lot of time and money some organizations state that they are unable to utilize its benefits that they intend to get by implementing ERP software [7, 9]. As stated by Hong and Kim [10], at least the three quarters of ERP projects are judged to be a failure.

Managers emphasize that ERP implementation projects are the most difficult system development projects [11]). Especially, ERP applications change procedures and processes of organizations into a software system. Due to complexity, time and workforce requirements, after implementation it nearly becomes impossible to rollback [8].

Due to the risks, before purchasing ERP software, it is necessary to determine if the ERP software bears on the ability to satisfy stated or implied needs of its customers by the use of Software Quality (SQ) models [12]. As stated in ISO 25000 [13] there is no general software classification system and the importance of quality characteristics for each type of software depends on the type of software. Requirement levels of software quality characteristics needs to be determined to evaluate particular type of software product, by the use of the Software Quality models. As an example, for web applications studies on determining software quality are performed by Calero and Olsina [15, 16]. Likewise, such a study is necessary for ERP software.

In this study, we determine requirement levels of software quality characteristics according to users' view of quality for ERP software. This set of quality characteristics and their requirement levels can be used as a checklist to evaluate if ERP software bears on the ability to satisfy stated or implied needs by ERP software purchasers. By the use of this checklist, it is possible to reduce the risk related to ERP software.

The remainder of this paper is organized as follows. Section 2 gives an overview of the associated literature on external software quality characteristics. Transactional backbone of ERP systems is reviewed in Section 3. In Section 4, a checklist including requirement levels of ERP software according to the user view is given and some formulas are introduced for the purpose of evaluating particular ERP software. In the last section of the paper, summary and conclusion are presented.

2 External Software Quality and Its Characteristics

The term quality is defined by Crosby [14] as "conformance of requirements". Feigenbaum [17] puts forward another definition of quality as total composite characteristics of a product or service to meet the expectations of customer while Juran [18] states the phrase "fitness for use". Although the quality definitions give guidance on definition of software quality, software quality can be described from five different perspectives [19]. These perspectives are transcendental view, user view, manufacturing view, product view and value based view. When these perspectives are evaluated a general definition for software quality can be described as follows: a total composite of characteristics of software product which bears on the ability to meet the stated or implied needs [13].

The definition of Software Quality shows that characteristics and requirements are the most important factors. Characteristics contribute to fulfilment of requirements

and software quality arises from characteristics which are appropriate to the requirements [20]. The relationship between these factors and software quality is given in Figure 1.

Fig. 1. Relationship between characteristics, requirements and software product [18]

One of the earliest studies that tried to provide a framework for Software Quality and its characteristics was proposed by McCall et.al. [21]. Another study is put forward to constitute a set of factors that affect Software Quality by Boehm et.al. [22]. Bowen et.al. [23] are proposed with a larger number of characteristics after few years. As seen below, these early studies try to constitute software quality model by providing a set of characteristics.

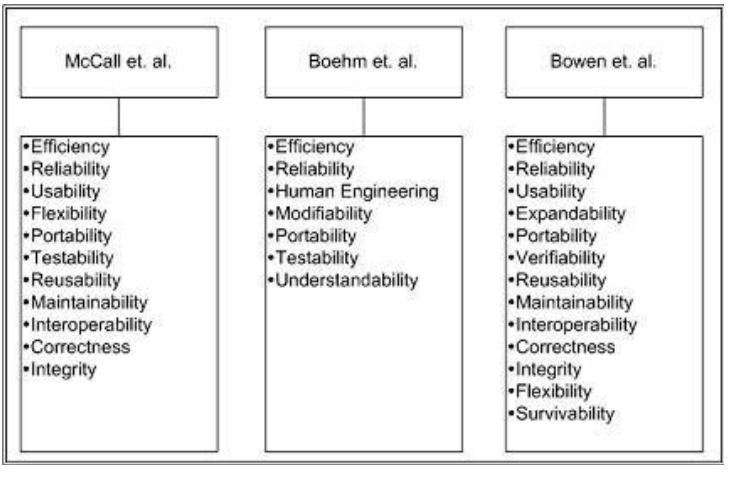

Fig. 2. Early software quality models

There are a lot of subsequent attempts to provide a software quality model [24-28]. Although, these sets of quality models including early models seem to cover the same identical characteristics, the definitions of these characteristics are different. ISO 25000 [12] software quality model is developed to establish international agreement and it is developed further with four parts by ISO [13, 29-31]. The international standard covers six quality characteristics which are shown in Figure 3 [13].

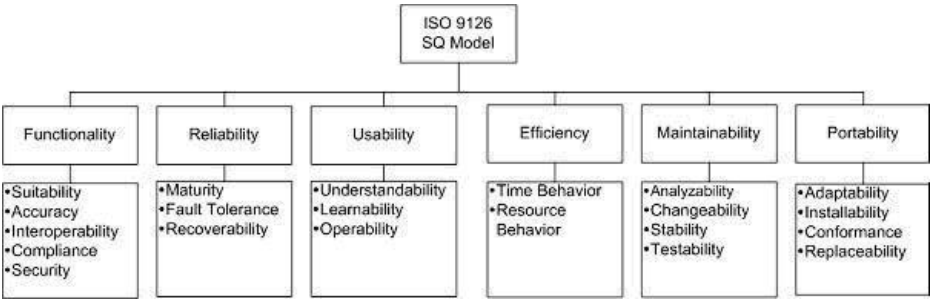

Fig. 3. ISO 25000 Software Quality Model [13]

Although international standard consolidated many different views of quality for software, there are still some other views that are not included into ISO 25000 and some of them cover more characteristics. In his model, Dromey [32] puts forward one more software quality characteristic, reusability. The other most significant attempt for constituting a set of characteristics is put forward by Software Engineering Committee [33]. Their model includes two more characteristics. These additional characteristics and sub-characteristics are given in Figure 4.

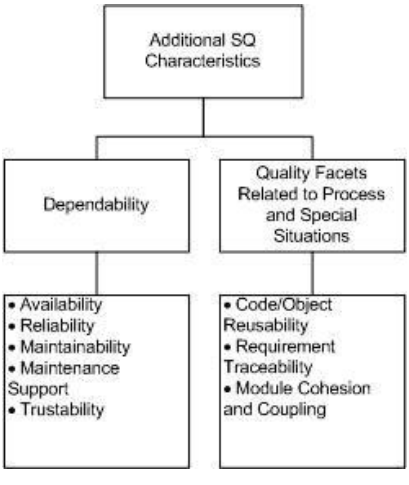

Fig. 4. Additional software quality characteristics [47, 48]

As it is can be understood from these, there are some objection against international standard. However, debate on this issue is settled; since, there is no additional study after this standard. Therefore, we used software quality model of the standard.

As it is obviously seen in the definitions of quality, meeting the requirements of users is stressed. Likewise, Software Quality is also highly related to conforming to the requirements of end-users [34]. Moreover, proposed study aims to provide a guide

to ERP purchasers (i.e. end-users) to evaluate software products before implementing them. Therefore, External Software Quality Characteristics need to be explained.

As it is stated in the standard the relationship between software quality and its characteristics depends on the type of the user. The end-users are concerned with functionality, reliability, usability and efficiency characteristics [29]. These characteristics are called as External Software Quality Characteristics.

Definitions of these characteristics and sub-characteristics are clearly identified in the standard [29]. Since the study is based on external metrics, descriptions on characteristics are given in Table 1 and descriptions on sub-characteristics are given in Table 2.

Table 1. Descriptions of External Software Quality Characteristics [29]

Characteristics	Description
Functionality	The capability of the software to provide functions which meet stated and implied needs when the software is used under specified conditions.
Reliability	The capability of the software to maintain the level of performance of the system when used under specified conditions.
Usability	The capability of the software to be understood, learned, used and liked by the user, when used under specified conditions.
Efficiency	The capability of the software to provide the required performance relative to the amount of resources used, under stated conditions.

Table 2. Descriptions of External Software Quality Sub-Characteristics [29]

Characteristics	Sub-Characteristics	Description
Functionality	Suitability	Attribute of software that bears on the presence and appropriateness of a set of functions for specified tasks.
	Accuracy	Attributes of software that bear on the provision of right or agreed results or effects.
	Interoperability	Attributes of software that bear on its ability to interact with specified systems.
	Compliance	Attributes of software that make the software adhere to application-related standards or conventions or regulations in laws and similar prescriptions.
	Security	Attributes of software that bears on its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.

Table 2. (*continued*)

Reliability	Maturity	Attributes of software that bear on the frequency of failure by faults in the software.
	Fault Tolerance	Attributes of software that bear on its ability to maintain a specified level of performance in cases of software faults or of infringement of its specified interface.
	Recoverability	Attributes of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it.
Usability	Understandability	Attributes of software that bear on the users' effort for recognizing the logical concept and its applicability.
	Learnability	Attributes of software that bear on the users' effort for learning its application.
	Operability	Attributes of software that bear on the users' effort for operation and operation control.
Efficiency	Time Behaviour	Attributes of software that bear on response and processing times and on throughput rates in performing its function.
	Resource Behaviour	Attributes of software that bear on the amount of resources used and the duration of such use in performing its function.

3 Enterprise Resource Planning Systems

In this section of the study literature review on ERP systems is given; since, it is aimed to weight (i.e. determine the requirement level) each of the External Software Quality Characteristics and Sub-Characteristics to evaluate ERP. ERP systems usually cover a technical infrastructure, transactional backbone and advanced applications [35]. ERP systems can roughly be described as software that integrates distributed applications of finance, human resources, production, sales, purchase, supply and distribution [36].

Stated or implied expectations of production systems to utilize ERP systems can be found by elaborating on the chronological development processes of these systems. By this approach user expectations can be better understood.

It dates back to 1960s that the manufacturing systems first discovered Material Requirement Planning (MRP) [36]. MRP was the most effective tool for improving operations by calculating material requirements and requirement periods at that time

[37]. In the beginning of 1980s change in the market conditions caused manufacturing firms to seek innovative techniques. By adding new procedures to MRP, Manufacturing Resources Planning (MRP II) developed. MRP II try to integrate MRP and some other functional operation areas like marketing and finance [38]. Moreover; MRP II does not only cover MRP but also plans capacity [39].

After MRP II, Distribution Requirement Planning (DRP) and Computer Integrated Manufacturing (CIM) emerge. By using DRP, it becomes possible to plan and manage distribution channels and product deliveries [40]. CIM covers the applications of integration of the manufacturing processes and technical functions [3]. A few years after CIM and DRP, a new way of planning all of the resources of any organization Enterprise Resource Planning is born [35].

Explicit reason for evolution of ERP systems is to integrate former applications [6]. On the other hand, it must be noted that this is not the only reason that gives birth to ERP systems. In the last century small corporations changed into modern and global enterprises. Moreover, in the last three decades market conditions changed so quickly that those enterprises face hard competition [2]. Shortening of life cycle of products and rapid increase in the product diversity are the most significant factors all of which are shown to put enterprises under a big pressure [3].

Because of these conditions, production systems try to find a systemic idea involving all aspects of resource planning. They start to seek a way to restructure their processes flexibly. Most important factors for seeking this kind of integration is to enable production systems to deliver higher variety of products at lower cost, supply customized products, develop new production strategies focusing on individual customers and plan all of the resources more effectively [4].

It seemed possible to use IT in an integrated way to move towards the creation of appropriate infrastructure [41]. Under these circumstances the idea of Enterprise Resource Planning (ERP), which is able to plan all of the resources of organizations in an integrated way, is born. The main point of view of developing such systems is restructuring organizations in a process oriented way rather than function oriented [6]. Difference of process and function oriented enterprise structure is shown in Figure 5.

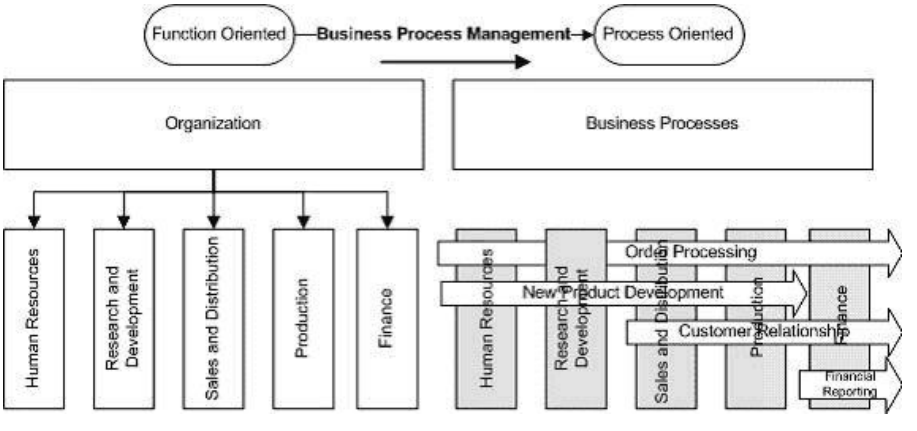

Fig. 5. Structures of process oriented and function oriented enterprises (Skok and Döringer 2001)

Today ERP systems are utilized as software applications. ERP systems can be defined in three different ways: [42] A commercial software product that is sold and purchased; (2) A management tool that held all of the data and processes of an organization; [42] A key factor that gives solutions to the infrastructure of business processes [43]. In this study, the focus will be on an ERP system as a software product; since, the primary goal is to evaluate quality of ERP software.

The term "enterprise" refers to every function of a system that supplies services and/or products. ERP software provides one database, one application, and one user interface for distributed functions of enterprises [44]. All properties of ERP systems are shown in Figure 6.

These distributed functions cover production planning, purchase, inventory control, finance, human resources and sales and marketing [45]. In the last decade, Customer Relationship Management (CRM), Supply Chain Management and web applications are added on ERP systems as advanced applications [35].

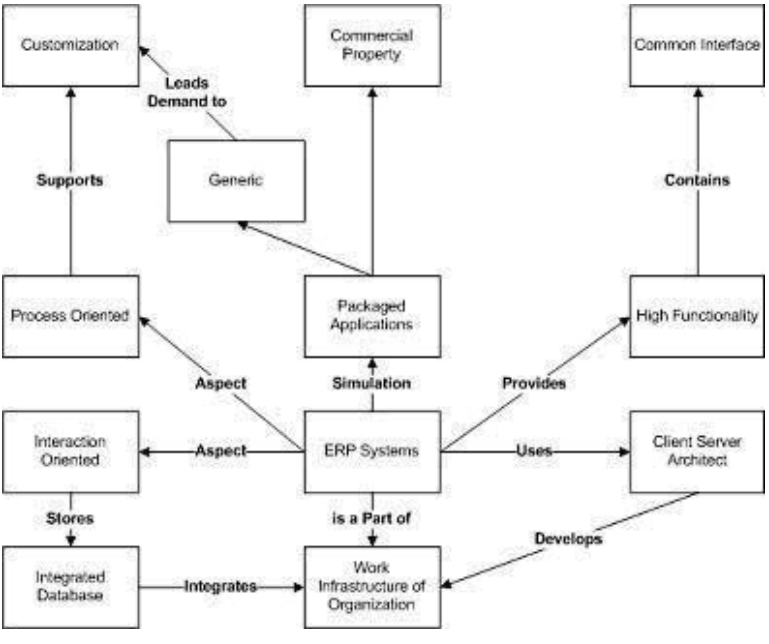

Fig. 6. Fundamental properties of ERP software [49]

Shields (2001) gives a framework for an eXtended Enterprise Systems (XES) and he defines XES as a systems that covers managerial portal, data warehouse, advanced applications, transactional backbone of ERP and a technical infrastructure. In reality ERP systems as software products also works on a technical infrastructure and those technical infrastructures include hardware, network, database management system, e-mail and gateway. On the other hand, it must be emphasized that in this study advanced applications and technical infrastructure are not going to be considered as a

part of ERP systems' core. In this study, we focus on transactional backbone of ERP systems; since, there are a lot of different advanced applications [35].

Definition of Shields [35] gives a good guidance for determining core of ERP systems. Following his guidance, it can be stated that the finance, sales and distribution, human resources and manufacturing components constitute transactional backbone of ERP [7, 35]. Visual representation of the transactional backbone of ERP systems is given in Figure 7.

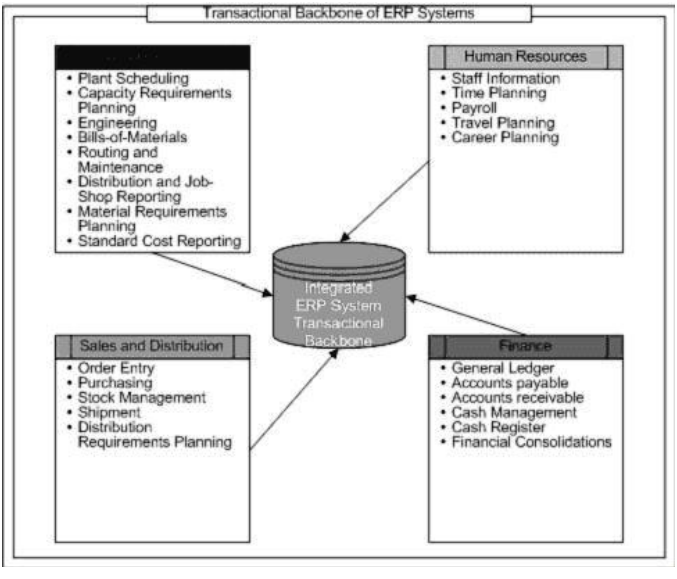

Fig. 7. Transactional backbone of ERP systems

It must be noted that ERP systems as a software product is not limited with the given components. Like the other software products, ERP systems also have security, authorization and help components and human-computer interfaces [43]. Moreover, these components are equally important for the users of software products. Therefore, while evaluating external quality of ERP, not only transactional backbone but also these additional components (security, authorization and help) must be considered.

4 Determining External Quality of ERP Software

It is obvious that the idea of ERP can satisfy the requirements of production systems. Although the idea of ERP systems is unique; ERP software products have many common features and also have significant differences. Therefore, end-users must consider related risk and evaluate these systems by considering conformance of particular software to their stated or implied needs.

Today, measuring External Quality Characteristics of a software product is considered as a valuable tool in Software Quality evaluation to determine conformance to the requirements of its users. In the beginning, External Software Quality

Characteristics can help them to examine software according to each Sub-Characteristic. In the frame of this reference, before buying any software product purchasers can test software and evaluate it according to external metrics.

When evaluating external metrics, end-users can grade software according to External Software Quality Sub-Characteristics. This grading can be a number between 0 and 100 as it is recommended in the standard. If this grade is denominated as Eij, it can be explained as follows:

Eij: Evaluation grade assigned by end-users to a Sub-Characteristic for software
where
i: External Software Quality Characteristic (i= 1, 2, 3, 4)
j: External Software Quality Sub-Characteristic (j= 1,… J)

On the other hand, while comparing different ERP software products it is hard to determine which ERP software best satisfies their requirements; because, each Quality Characteristic and Sub-Characteristic has different weights. With this respect, the present study tries to provide a guide by setting up a checklist of weighted Quality Characteristics to be used by organizations and/or users that intend to purchase ERP software. These external quality Sub-Characteristics are weighted according to requirement levels when considering core of ERP software. Afterwards, these weights and examination results of end-users can be used as a tool for determining External Quality of any ERP software.

To determine requirement levels, first a questionnaire is set up according to the formerly given External Software Quality Characteristics and Sub-Characteristics. Questions are set up in such manner that each one indicates a significant Software Quality Sub-Characteristic. Subsequently, five expert professors all of whose research areas of interest is Information Technology, Production Systems and ERP Systems are chosen to answer the questionnaire. While preparing questionnaire the following rules are taken as a guide:

1. The aim of the questionnaire is presented, a brief explanation is given and formerly defined transactional backbone of ERP software is elaborated
2. It is tried to be comprehensible while asking questions. Explanations are kept precise and brief.
3. It is tried to keep questionnaire short and attractive. It is tried not to reference to a previous question in any questions.
4. It is tried to structure the questionnaire well and experts are expected to follow its logical order.
5. Answers to the questions are pre-defined and each interviewee is expected to assign a value from 1 to 10 for each question which refers to a particular Sub-Characteristic. These values refer to a degree of requirement where 1 refers to lowest degree of requirement and 10 refer to highest degree of requirement. While assigning those values experts are expected to assign a high value to a question that is considered of high importance.
6. It is tried to keep questions in such a manner that they do not affect judgements of experts.
7. The experts are not allowed to impose their opinions by leading to specific answers while performing questionnaire.

Answers to these questions give weights of each external software quality sub-characteristics since each question indicates requirement levels of a significant External Software Quality Sub-Characteristic. The results are given in Table 3.

Table 3. Weights of each External Software Quality Sub-Characteristics

Characteristics (i)	Sub-Characteristics (j)	Expert 1	Expert 2	Expert 3	Expert 4	Expert 5	Wij	Dij	Wi	Di
1. Functionality	1. Suitability	10	10	10	9	10	9,8	0,212	9,240	0,281
	2. Accuracy	10	10	10	9	10	9,8	0,212		
	3. Interoperability	8	10	10	9	9	9,2	0,199		
	4. Compliance	9	10	8	8	9	8,8	0,190		
	5. Security	9	10	8	8	8	8,6	0,186		
2. Reliability	1. Maturity	10	8	7	7	8	8,0	0,320	8,333	0,253
	2. Fault Tolerance	8	10	9	9	7	8,6	0,344		
	3. Recoverability	9	10	10	6	7	8,4	0,336		
3. Usability	1. Understandability	7	7	7	8	7	7,2	0,327	7,333	0,223
	2. Learnability	7	8	8	8	7	7,6	0,345		
	3. Operability	6	8	8	8	6	7,2	0,327		
4. Efficiency	1. Time Behaviour	8	8	9	8	9	8,4	0,525	8,000	0,243
	2. Resource Behaviour	7	8	8	8	7	7,6	0,475		

Notation for the formula below is explained subsequently:

Wij: Weight of each Sub-Characteristic
Dij: Relativistic degree of importance of each Sub-Characteristic of one Characteristic
Wi: Weight of each Characteristic
Di: Relativistic degree of importance of each Characteristic among all Characteristics

where

i: External Software Quality Characteristic (i= 1, 2, 3, 4)
j: External Software Quality Sub-Characteristic (j= 1,… J)
Weight of each Sub-Characteristic [46] is the average of grades assigned by experts as seen in the formula. Therefore, weights of each Sub-Characteristic can be calculated by the following equation.

$$W_{ij} = \frac{\sum_{k=1}^{K} G_{kij}}{K}$$

where
Gkij: Grade assigned by the kth expert for the jth Sub-Characteristic of the ith Characteristics
k: Number of expert (k= 1,…5)
As it is seen in the table each Sub-Characteristics is related to one Characteristic relatively more important than one another. Therefore, to compare each Sub-Characteristics of one Characteristics relativistic degree of importance for each Sub-Characteristic can be calculated by using below formula.

$$D_{ij} = \frac{W_{ij}}{\sum_{j=1}^{J} W_{ij}}$$

where
Dij: Relativistic degree of importance of a Sub-Characteristic for the ith Characteristic
By calculating average of Sub-Characteristics related to a Characteristic, weights of Characteristics can be found. For this purpose following equation can be utilized.

$$W_i = \frac{\sum_{j=1}^{J} W_{ij}}{J}$$

Moreover, relativistic degree of importance of each Characteristic can be calculated by using following formula.

$$D_i = \frac{W_i}{\sum_{i=1}^{4} W_i}$$

where
Wi: Weight of the ith Characteristic.
Di: Relativistic degree of importance of the ith Characteristic

As an example, the relativistic degree of importance of suitability Sub-Characteristic which is related to functionality is 21.2%. To give one more example it can be inferred that the efficiency quality characteristic has 24.3% effect on overall external quality when considering requirements of systems that intends to use ERP software.

This paper recommends ERP software buyers to use these weights before purchasing ERP software to find best alternative which meets their stated or implied needs. By using requirement levels, these weights help to evaluate External Software Quality of ERP software. For this purpose following equation can be used.

$$ESQoERP = \sum_{i}^{4} \sum_{j}^{J} W_{ij} * E_{ij}$$

where
ESQoERP: Weighted External Software Quality of a particular ERP software

The given formula can be used to evaluate different ERP software products. If this approach is followed and alternatives are graded, selection can be made by choosing the software which has highest grade.

As it is stated before, transactional backbone of ERP software covers different components. Moreover, experts that contributed to this study recommended using one another variable; since, each module has different degree of importance dependent on the significant differences of the production systems. Especially, manufacturing and service systems have major differences. Therefore, in this study it is proposed to assign one more weight to each component for manufacturing and service systems. With this respect, in Table 4 weights of components for those systems are proposed.

Table 4. Proposed weights of components for manufacturing systems and service systems

Components (C)	Manufacturing Systems	Service Systems
1. Sales and Distribution Components	0.20	0.20
2. Manufacturing Components	0.30	0.25
3. Human Resources Components	0.15	0.20
4. Finance Components	0.25	0.25
5. Additional Components	0.10	0.10

If it is desired to weight each module individually by using these weights component based weighted ERP software product quality can be calculated by the subsequent formula.

$$CbESQoERP = \sum_{l}^{5} C_l * \sum_{i}^{4} \sum_{j}^{J} W_{ij} * E_{ij}$$

where
CbESQoERP: Component based weighted External Software Quality of particular ERP software
Cl: Weight of lth Component
l: Number of component (l= 1,…5)

In some cases due to special conditions according to the requirements of systems evaluators, different weights than proposed ones can be used. Under such circumstances, by following the below constraint evaluators can define their weights.

$$\sum_{l}^{5} C_l = 1$$

Finally, by using the given formulas a particular ERP software product's overall external quality can be calculated. These evaluations will help the users to systematically evaluate ERP software and will give chance to benchmark different products.

5 Conclusion

In the field of IT, the most of the actions can be rolled back by its administrator or user. Only exception is implementation of ERP software within production systems. Purchasers of these products explore better ways to evaluate quality of these software products before making decision on buying one of them. With this respect, it is tried to determine the requirement levels of ERP software and weight them accordingly. By providing equations it is tried to provide a guide to make it possible to benchmark different ERP software products in a systematic manner. Even though proposed approach can be applied to any other software products, in this study ERP software is chosen; since, risk related to ERP software is more than other software.

In its current form, the present study proposes a model for evaluating external software quality of ERP software. In the near future, major ERP software can be evaluated in a number of production systems and compare the results to figure out the effectiveness of the model proposed. In addition to these, while using ERP software quality in use can become an important issue. Therefore, the main limitation of the current study is that it does not incorporate quality in use. In the near future, model for evaluating quality in use can be developed for ERP software.

References

1. Goldhard, J.: Business Strategies for the 21st Century Manufacturing Firm-Using CIM for Competitive Advantage. In: AUTOFACT, USA (1992)
2. Sohal, A.: A Longitudinal Study of Planning and Implementation of Advanced Manufacturing Technologies. International Journal of Computer Integrated Manufacturing 10(1-4), 81–95 (1997)
3. Nagalingam, S.V., Lin, G.C.I.: Latest Developments in CIM. Robotics and Computer Integrated Manufacturing 15, 423–430 (1999)
4. Da Silveria, G., Borenstein, D., Fogliatto, F.: Mass Customization: Literature Review and Research Directions. International Journal of Production Economics 72, 1–13 (2001)
5. Digre, T.: Bussiness Application Components. In: Object Oriented Programming Systems Languages Application, Austin, USA (1995)
6. Skok, W., Döringer, H.: Potential Impact of Cultural Differences on Enterprise Resources Planning (ERP) Projects. The Electronic Journal of Information Systems in Developing Countries 7(5), 1–8 (2001)

7. Davenport, T.H.: Putting the Enterprise into the Enterprise System. Harvard Business Review 76(4), 121–132 (1998)
8. Kumar, V., Maheshwari, B., Kumar, U.: An Investigation of Critical Management Issues in ERP Implementation: Emprical Evidence from Canadian Organizations. Technovation (2001)
9. Markus, M.L., Tanis, C.: The Enterprise System Experience: From Adoption to Success. In: Framing the Domains of IT Management: Projecting the Future through the Past. Pinnaflex Educational Resources Inc., Cincinnati (2000)
10. Hong, K.K., Kim, Y.G.: The Critical Success Factors for ERP Implementation: An Organizational Fit Perspective. Information & Management (40), 25–40 (2002)
11. Wider, C., Davis, B.: False Starts, Strong Finishes. Information week 711, 41–53 (1998)
12. ISO, ISO 25000: Quality Software Requirements and Evaluation. The International Organization for Standardization, Geneva (2004)
13. ISO, ISO/IEC 9126: Product Quality - Part 1: Quality Model (2001)
14. Crosby, P.: Quality is Free. McGraw-Hill, New York (1979)
15. Calero, C.: Handbook of Research on Web Information Systems Quality (2008)
16. Olsina, L.: Measuring Web Application Quality with WebQEM. IEEE Multimedia (2002)
17. Feigenbaum, A.V.: Total Quality Control. McGraw-Hill, New York (1991)
18. Juran, J.: Juran on Quality by Design. The Free Press, New York (1992)
19. Xenos, M., Christodoulakis, D.: Evaluating Software Quality by the Use of User Satisfaction Measurements. In: The 4th Software Quality Conference, University of Abertay Dundee (1995)
20. Xenos, M.: Usability Perspective in Software Quality. In: Usability Engineering Workshop, The 8th Panhellenic Conference on Informatics with International Participation, Southern Cyprus (2001)
21. McCall, J.A., Richards, P.K., Walters, G.F.: Factors in Software Quality. Rome Air Development Centre, Rome (1977)
22. Boehm, B.W.: Characteristics of Software Quality. North Holland Publishing Co, New York (1978)
23. Bowen, T.P., Wigle, G.B., Tsai, J.T.: Specification of Software Quality Attributes. Rome Air Development Centre, Rome (1985)
24. Grady, R.B., Caswell, D.L.: Software Metrics: Establishing a Company-Wide Program. Prentice-Hall, London (1987)
25. Deutsch, M.S., Willis, R.R.: Software Quality Engineering. Prentice-Hall, London (1988)
26. Forse, T.: Qualimetrie des Systems Complexes. Les Editions d'Organisation (1989)
27. Von Maryhauser, A.: Software Engineering Methods and Management. Academic Press (1990)
28. Khoshgoftaar, T.M., Allen, E.B.: Classification Techniques for Predicting Software Quality: Lessons Learned. In: Annual Oregon Workshop on Software Metrics, University of Idaho, USA (1997)
29. ISO, ISO 9126: Product Quality - Part 2: External Metrics (2003)
30. ISO, ISO 9126: Product Quality - Part 3: Internal Metrics (2003)
31. ISO, ISO 9126: Product Quality - Part 4: Quality in Use Metrics (2004)
32. Dromey, R.G.: Software Product Quality: Theory, Model and Practice, Software Quality Institute, Brisbane, Australia (1998)
33. SEC, Knowledge Area: Software Quality Analysis, The Software Engineering Body of Knowledge (SWEBOK), Software Engineering Committee, Institute of Electrical and Electronics Engineers, Inc., Montreal (1999)

34. Stavrinoudis, D.: Early Estimation of Users' Perception of Software Quality. Software Quality Journal 13, 155–175 (2005)
35. Shields, M.G.: E-Business and ERP - Rapid Implementation and Project Planning. John Wiley & Sons, Inc., New York (2001)
36. Light, B., Holland, C.: Enterprise Resource Planning Systems: Impacts and Future Directions. In: Systems Engineering for Business Process Change: Collected Papers from the EPSRC Research Programme. Springer, London (2000)
37. Plenert, G.: Focusing Material Requirements Planning (MRP) towards Performance. European Journal of Operational Research 119, 91–99 (1999)
38. Browne, J., Harhen, J., Shirman, J.: Production Management Systems. Addison-Wesley (1988)
39. Hatzilygeroudis, I.: MRP II-Based Production Management Using Intelligent Decision Making. In: Beyound Manufacturing Resource Planning. Advanced Models and Methods for Production Planning. Springer (1998)
40. Greene, J.: Production and Inventory Control Handbook. McGraw-Hill (1987)
41. Hussein, J.: Providing an Insight on Improving Performance of MRP. Clemson University, Clemson (2000)
42. Hammer, M., Champy, J.: Reengineering the Corporation: A Manifesto for Business Revolution. Nicholas Brearley Publishing, London (1993)
43. Alageo, M.E.A., Barkmeyer, E.J.: An Overview of Enterprise Resource Planning Systems in Manufacturing Enterprises, National Institute of Standards and Technology (1999)
44. Cambashi, Enterprise Resources Planning for Manufacturers. Cambashi Ltd., Cambridge (1999)
45. Brislen, P., Krishnakumar, K.: What is ERP, Enterprise Resource Planning (1999)
46. Bertrand, J.W.M., Zuijderwijk, M., Hegge, H.M.H.: Using Hierarchical Psuedo Bills of Material for Customer Order Acceptance and Optimal Material Replenishment in Assemble to Order Manufacturing of Non-Modular Products. In: International Journal of Production Economics (2000)
47. Software Engineering Committee, Knowledge Area: Software Quality Analysis, The Software Engineering Body of Knowledge (SWEBOK). Institute of Electrical and Electronics Engineers, Inc., Montreal (1999)
48. Dromey, R.G.: A Model for Software Product Quality. IEEE Transactions on Software Engineering (1995)
49. Hagman, A.: What will be of ERP? Could Component Software Spell a Strategic Inflection Point for the Industry?, School of Information Systems Queensland University of Technology, Queensland (2000)