Czech University of Life Sciences Prague
Faculty of Economics and Management
Department of System Engineering

Diploma Thesis

Comparison of the two software products quality for accounting used in Russia and in the Czech Republic

Vasily STEPANOV

The official document

Declaration

I hereby declare that I have worked on my Diploma thesis titled "Comparison of the two software products quality for accounting used in Russia and in the Czech Republic" solely and completely on my own and that I have marked all quotations in the text. The literature and other material I have used are mentioned in the References section of the thesis.

In Prague on 7th April 2011

____

Vasily Stepanov

Acknowledgement

First of all I would like to express my sincere gratitude to my supervisor Prof. RNDr. Jiří Vaníček, CSc. for my supervision and guidance. He gave me valuable suggestions and made corrections. Under his guidance I overcame many difficulties and learn a lot.
It is a pleasure to thank those who made this thesis possible as Dr. Ing. Robert Pergl and Ing. Tomáš Rain PhD. They had made available his support in a number of ways.
I thank all teachers from our Faculty of Economics and Management for interesting lectures, knowledge and skills that they provided me.
I am grateful to my colleagues and friends. They inspired me to do this research and helped with the challenges I faced in writing my thesis. I want to express special gratitude to Darima Batorova. She translated a part of my thesis to Czech language.
In addition, I would also like thank The International Relationship Office for their guidance and project "Erasmus Mundus" for the opportunity to study at the Czech University of Life Sciences and for the financial support.
Finally, I would like to pay high regards to my parents Maria and Innokenty, brother Ilya and my relatives for their sincere inspiration and encouragement throughout my research work.

Porovnání kvality dvou softwarových producktů pro účetnictvi užívaných v Rusku a Českě republice

Comparison of the two software products quality for accounting used in Russia and in the Czech Republic

Summary

For the last three years number of accounting software products rapidly increased and as the result selection process became more sophisticated. Therefore, quality level of each software product should be measured in order to make selecting procedure easier. This research presents quality evaluation and comparison on the case study of two accounting software products widely used in Russia and in the Czech Republic. For these purposes six quality models are explored and ISO 25010 Quality model is selected. Information for the quality rating is collected from the following sources: software development companies, software documentations and interviews with the users of the products. For the purpose of evaluation eight quality characteristics are introduced. Rating scale from zero to one is presented, where 0 stands for the worst and 1 stands for the best results. A fuzzy simple additive weighting method is used in order to obtain final rates. As a result five out of eight quality measures are more suitable for the Russian software than for the Czech one. According to the conducted evaluation the final rate for the Russian software product is equal to 0,593 and for the Czech software product is equal to 0,407. It indicates that the Russian software product has higher level of quality than the Czech one.

Keywords: ISO Standards, Software Quality, Software Quality Models, Evaluation, Quality measures.

Souhrn

Za poslední tři roky počet účetnického softwaru na trhu značně vzrostlo a následně se zjevila potřeba více propracovat proces výběru produktů. Pro zlepšení procedury výběru je nutné měřit úroveň kvality každého produktu. Tento výzkum popisuje možnosti hodnocení kvality produktů, které jsou otestované pomocí srovnávání dvou příkladů softwarových produktů. Vybrané účetnické softwarové produkty jsou široko používané v Rusku a v České Republice. Pro splnění cílu diplomové práce bylo prozkoumáno šest modelů požadavek na kvalitu a standard ISO 25010. Zdroji informací pro tento výzkum byly výrobci softwarových produktů, oficiální dokumentace, a také rozhovory s bezprostředními uživateli účetnického softwaru. Pro hodnocení bylo použito osm charakteristik. Posuzovací stupnice se počítala od 0 do 1, přičemž „0“ bylo nejhorší hodnocení a „1“ představovalo nejlepší výsledek. Finální závěr byl zpracován pomocí aditivní metody a konkrétně pomocí „fuzzy simple additive weighting method”. Na základě testování lze posoudit, že pět z osmi charakteristik kvality jsou více příslušné pro ruský software než pro české analogy. Ruský produkt dostal 0,593 a český softwarový produkt jen 0,407. Výsledky finální evaluace svědčí o lepší kvalitě ruského účetnického softwaru.

Klíčová slova: ISO standarty, jakosti softwaru, softwaru kvalitní modely, hodnocení, opatření pro kvalitu.

Content

Figures

Tables

1. Introduction

“When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind.”

(Lord Kelvin, Popular Lectures and Addresses, 1889)

Nowadays, there are many different accounting software products specializes in solving business problems. Level of competition between software development companies has become higher. Therefore quality requirements have become more sophisticated as well in order to protect users from unqualified software products. Small, medium and large sized business companies are faced the challenge selecting accounting software which will improve their business. Selection process includes many different aspects. Usually companies take into account such aspects as price, software functionality, hardware requirements and etc.

Selecting procedure of software product which perfectly fits business needs is associated with the quality term. Interested party should know how to measure quality of software products. It is possible to compare software products between each other only after measuring quality of each software product separately.

Traditional way of comparing software products with similar functionality is based on comparisons of prices and degree of its brand name recognition. Both of these approaches are not correct and can be cause of making wrong decision. As a result wrong decision will have a negative effect on company’s business. Software development companies are used various types of advertisements: TV, radio, internet, ads in magazines, seminars, demonstration shows in order to attract attention of potential customers. However the costs spending for advertising have become a part of software product price. Price is not a part of the quality; nevertheless is one of the criteria to the management decisions. Brand name is not indicates a high level of quality, but it indicates identity of a specific product.

There are many definitions for the term quality. In this paper quality declared as the degree to which a set of inherent characteristics fulfils the requirements. [VANÍČEK, 2010]

There are lots of standards which had been established by International Organization for Standardization and by International Electromechanical Commission in order to protect users from low-quality products. These two organizations found join technical committee – ISO/IEC JTC1 Information technology. [VANÍČEK, 2010]

For evaluation of software quality had been established lots of different standards and quality models. ISO standards will be presented in details. In this paper will be introduced and discussed different methods and techniques for evaluation of software products quality, criteria's for right decision making as well.

The paper is organized as follows. Chapter 2 presents objectives and methodology. Chapter 3 presents overview of ISO standards, definitions of information systems and software products quality according to ISO 25010 standard. Alongside with ISO standards six quality models are introduced. Chapter 4 discusses the case study of two accounting software products used in Russia and in the Czech Republic. Evaluation of software products quality is examined according to the ISO 25010 methodology.

2. Objectives and methodology

The aims of research in this paper are:

- To get acquainted with ISO standards
- To get acquainted with existing Quality Models
- To evaluate software products quality according to the ISO 25010 standard
- To compare quality for the accounting software products according to a Fuzzy Simple Additive Weighting method (SAW)

Nowadays accountancy is very important term which can be bring up an association with the term “successful business”. It is very powerful instrument for business administration. Quite often companies deals with the problem: how to select “perfect” accounting software system which will fit all their business needs. Wrong selected accounting software system could be a reason of serious failures in the business administration.

Many different companies are offer software for accountancy. Also there are many articles and demonstration videos which are available in magazines and internet. The problem is that it is very difficult to identify what is missing and what is wrong with the software product. Weaknesses and negative points usually are skipped while video demonstration, overviews and presentations of the software product.

Nevertheless it is possible to analyze quality of software product impartially using methodology which is described in ISO Quality standards.

For evaluation of software quality six quality models are introduced: McCall’s, Boehm’s, Dromey’s, FURPS, ISO 9126, ISO 25010. One of the quality models is selected for evaluation process and introduced in details.

Two accounting software products widely used in Russia and in the Czech Republic are presents in order to compare their quality characteristics and rating.

Russian software product “1C Accounting 8” has been selected due to several reasons: “1C” Company has leadership position in software area for accounting in Russia. During my study in Russia I worked in one of the distribution centers “1C” Company. At that time I learnt functionality, system requirements for this software product.

The Czech software product “Pohoda” has been selected because of its popularity level among the Czech software development companies.

List of requirements for software product is established according to the user’s needs.

Evaluation criteria are established for measurement quality characteristic and sub-characteristics.

Information about software products is collected for evaluation of software quality. Sources of information are web pages of software developments companies, software documentations, interviews with the users of the software, interviews with the employees of software development companies, testing of demo versions for both software products. Quality characteristics and sub-characteristics are measured according to the methodology and comparison table is formed. Each quality characteristic is pairwise compared between software products.

A fuzzy simple additive weighting method introduced in order to rating software products quality. First step is to build decision matrix according to evaluation criteria. Once a decision matrix has been built, the decision maker evaluated its various elements using concrete data for the elements of his own judgments about the elements relative meanings and importance. [DOMEOVÁ, 2010]

All data in the table is normalized and transformed to all benefit criteria. Next step is standardization according to the formula:

$a_{ij} = \frac{x_{ij} - D_j}{H_j - D_j}$; Where

D_j - is a column of minimum values;

H_j - is a column of maximum values;

x_{ij} - is a value of *i*-th variant under *j*-th criterion;

a_{ij} - is a standardized value of x_{ij}

Score method is used for construction of the weight vector. Quality characteristics rating obtained from quality rating table.

Finally, utility (total trade-off) calculation presents for rating software products: $u(a_i) = \sum_{j=1}^{k} v_j a_{ij}$; where v_j is a rating for each criteria.

Decision maker compares the results of evaluation process and makes decision which software product has higher level of quality.

3. Literature review

This chapter is focused on overview of existing ISO standards and methods for quality estimation. First of all evaluator should select one quality model which is more suitable for evaluated software according to different approaches. Overview of six quality models will be given in this paper.

3.1. ISO/IEC standards overview

This part of chapter is addressed to explain the importance of the ISO standards. It contains brief descriptions of ISO standards which are about information systems and software products quality as well.

Standardization is very important, because standards help to unite the view of the points at issue and create uniform rules. It facilitates to combine products of various suppliers and herewith support competitive environment. With growing globalization of human society grows an emphasis that the standardization was at widest. The standards, which would be only valid local, have only a sense there, when goes for specific situation, which are characteristics for this region and rebound for example their cultural specifics. Information technology hasn't this local specification. Important are the standards, which are accepted worldwide. [VANÍČEK, 2010]

Different users of the product or different participants on the process can have different needs and expectations. Quality for the user's point of view is degree of satisfaction stated or implied to user needs. But quite often it is not easy to transform "fuzzy" user's needs into precise and correct requirements because sometimes end user does not know his needs. [VANÍČEK, 2010]

Most of all people are acquainted with ISO 9000. This standard is about quality management. Unfortunately, it cannot be useful for quality measurement of specific information system or software product because it describes only general information about quality management systems and specifies the terminology.

In the area of information technology, there operate two main worldwide standardisation organizations: ISO and IEC. To avoid duplication of the standardization effort, the ISO and IEC have created the joint standardisation authority called the “ISO/IEC Joint technical committee for information technology” – “ISO/IEC JTC1 – Information technology”. The ISO/IEC JTC1 outputs are automatically assumed to be ISO and IEC documents and have the ISO/IEC label. [VANÍČEK, 2010]

There are several standards which describes quality. Project ISO/IEC SQuaRE 250xx (Software Quality Requirements and Evaluation) [ISO, 2010] replaced ISO/IEC 9126 series – Software product quality. ISO/IEC 9126 has been published in 1991. After ten years was published first technical report and within three years there were published three more technical reports.

1. ISO IS 9126-1: Quality model [ISO, 2001].
2. ISO TR 9126-2: External metrics [ISO, 2003].
3. ISO TR 9126-3: Internal metrics [ISO, 2003].
4. ISO TR 9126-4: Quality in Use metrics [ISO, 2004]

SQuaRE consists of the following five divisions:

Quality Management Division (2500n). The International Standards that form this division define all common models, terms and definitions further referred to by all other International Standards from the SQuaRE series. The division also provides requirements and guidance for a supporting function that is responsible for the management of the requirements, specification and evaluation of software product quality. [ISO, 2010]

Quality Model Division (2501n). The International Standards that form this division present detailed quality models for computer systems and software products, quality in use, and data. Practical guidance on the use of the quality models is also provided. [ISO, 2010]

Quality Measurement Division (2502n). The International Standards that form this division include a software product quality measurement reference model, mathematical definitions of quality measures, and practical guidance for their application. Examples are given of internal and external measures for software quality, and measures for quality in use.

Quality Measure Elements (QME) forming foundations for these measures are defined and presented.[ISO, 2010]

Quality Requirements Division (2503n). The International Standards that form this division help specify quality requirements, based on quality models and quality measures. These quality requirements can be used in the process of quality requirements elicitation for a software product to be developed or as input for an evaluation process. [ISO, 2010]

Quality Evaluation Division (2504n). The International Standards that form this division provide requirements, recommendations and guidelines for software product evaluation, whether performed by evaluators, acquirers or developers. The support for documenting a measure as an Evaluation Module is also present. [ISO, 2010]

SQuaRE Extension Division (ISO/IEC 25050 – ISO/IEC 25099). These International Standards currently include requirements for quality of Commercial Off-The-Shelf software and Common Industry Formats for usability reports. [ISO, 2010]

Figure 3-1 Organization of the SQuaRe series of standards (Source: ISO/IEC 25010, 2010)

Standard ISO/IEC 15939 Software Engineering – Software Measurement Process [ISO, 2007]. This standard does not only include quality, but generally measure of all attributes of software development and systems engineering. The standard is described in terms of the purpose and outcomes of a compliant process, along with associated activities and tasks. The standard also defines the measurement information model and associated terminology. The ISO/IEC 15939 covers measurement activities, required information, the application of measurement analysis results, and determining if analysis results are valid. The standard can be used by both systems suppliers and acquirers.

Another series of six standards, the ISO/IEC 14598-1 through ISO/IEC 14598-6 (1998, 1999, 2000, 2001) describes the quality of the evaluation process from different perspectives.

Part 1: General overview

Part 2: Planning and management

Part 3: Process for developers

Part 4: Process for acquirers

Part 5: Process for developers

Part 6: Documentation of evaluation modules.

Finally, the isolated standard ISO/IEC 12119 (1994) is focused on off-the-shelf distributed software (or, software packages). [VANÍČEK, 2010]

3.2. Information systems and software product quality

First of all let’s define properly what information system is and what software product is [VANÍČEK et al, 2010].

Information system: A system for collecting, storing, processing, making retrieval, and accessing of data and enabling the interpretation of information from data.

Information system consists of hardware, software, human staff, and organization rules.

Product: A result of process. There four generic product categories”

- Services.
- Software (SW) is more general concept than Computer software.
- Hardware (HW) is more general concept than Computer hardware.
- Processed materials.

In this document has been used next definition of software product: a software product is a set of programs, procedures, instructions and rules, perhaps with documentation and data, delivered to a third party under a single label. [VANÍČEK et al, 2010]

The main problem of software is that it cannot be used as such and consequently it cannot fulfil any needs or requirements. Software can only be used as a part of a system that usually also contains other software, computer hardware, operator services and operating procedures. Strictly speaking, the quality of a software product cannot be investigated and evaluated separately, but only within the context of complete system. [VANÍČEK et al, 2010]

Nevertheless, the impact of software on the quality of information and communication system as a complex entity is often crucial. For that reason, we consider the software product quality, as the impact of the software product on the quality of system in which it is integrated. However, the quality of a software product has to be measured and evaluated based on external behaviour of a complete system. The quality of any software component identifiable within the software product is evaluated in the same way. Then, the quality of a system can be considered as the result of the quality of individual components and their interaction. [VANÍČEK et al, 2010]

3.3. Quality model hierarchy

The SQuaRE quality model categorizes software quality into characteristics which are further subdivided into sub-characteristics and then into quality properties. [ISO,2010]

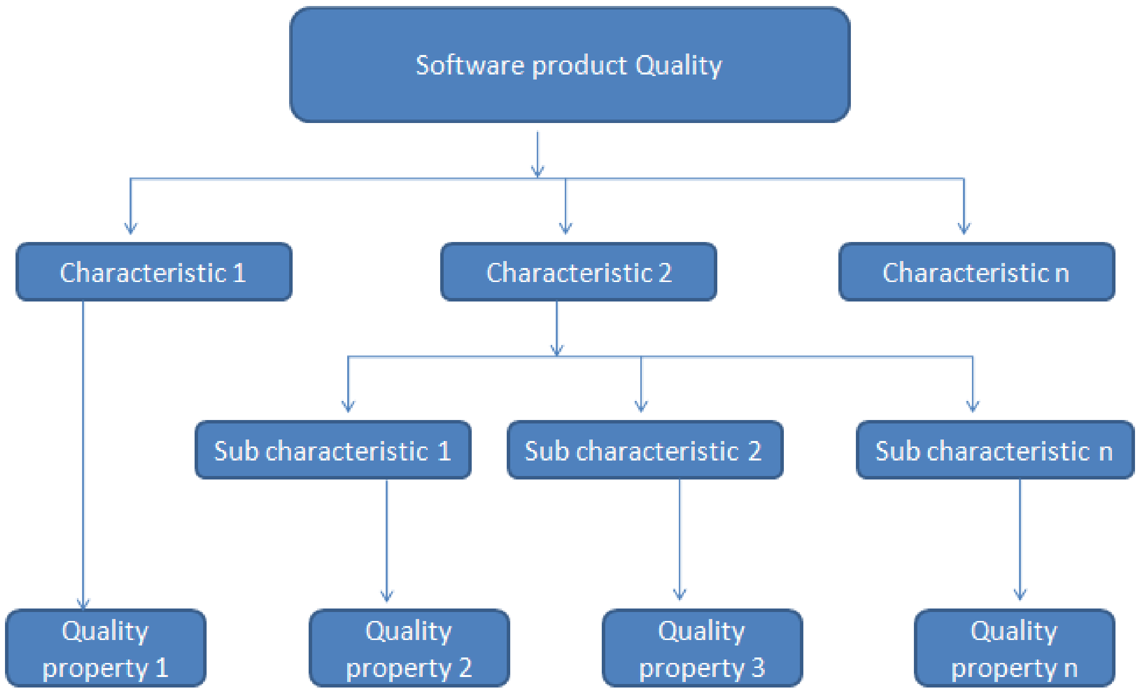


Figure 3-2 Structure used for the quality models (Source: ISO/IEC 25010, 2010)

The SQuaRe model consists of two parts, the model for External and Internal Software Quality and the model for Quality in use.

In reality the situation is more complicated. The intersection of two quality characteristics can be not empty. The same holds for the sub-characteristics intersection. One attribute can affect more than one sub-characteristics and also more than one characteristic. The real pasture of the quality characteristic – quality sub-characteristic – quality attributed and measures is presented on the Figure 3-3. [VANÍČEK, 2010]

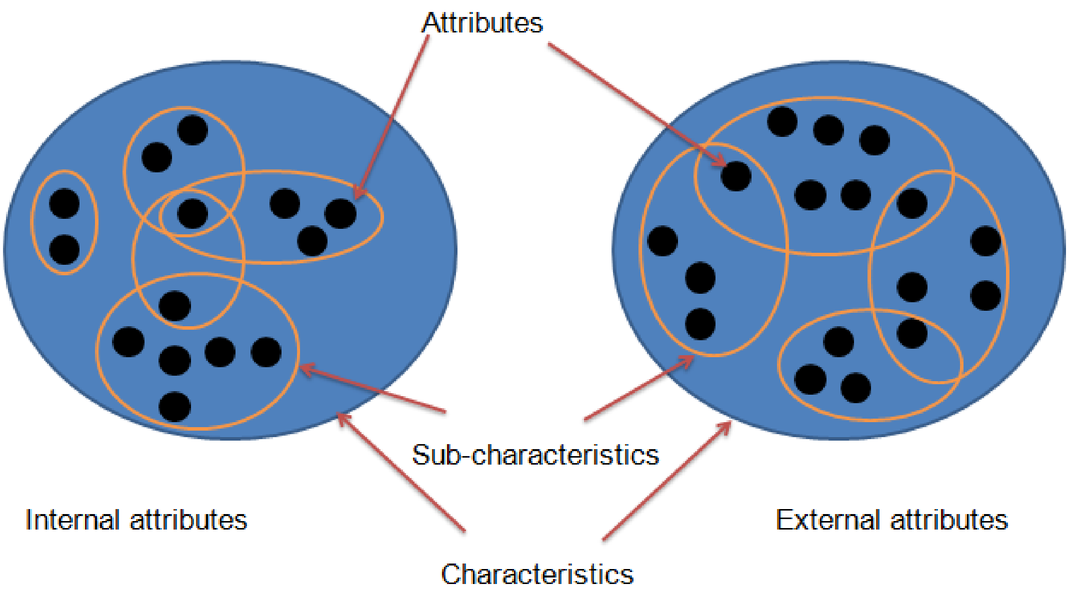


Figure 3-3 Real quality model hierarchy (Source: Vaníček Jiří, Information Systems. Quality Rating, 2010)

3.4. Quality measurement model

The software product quality measurement reference model describes the relationship between a quality model, its associated quality characteristics (and sub-characteristics), and the software product attributes with the corresponding software quality measures, measurement functions, quality measure elements, and measurement methods. Figure 3-4 shows that software quality measures are constructed by applying quality measure element also serves as a software quality measure, the measurement function applied would be identity function. [VANÍČEK, 2010]

Quality measure elements maybe either base or derived measures. Quality measures elements are constructed in accordance with the guidance provided in ISO/IEC15939.

Software quality measures are selected to satisfy the needs of developers, acquirers, managers, and others for information. In the context of the SQuaRE series, information needs may be defined by quality requirements and product quality evaluation. Criteria for selecting software quality measures and quality measure elements to fulfill those information needs shall be documented. [ISO, 2010]

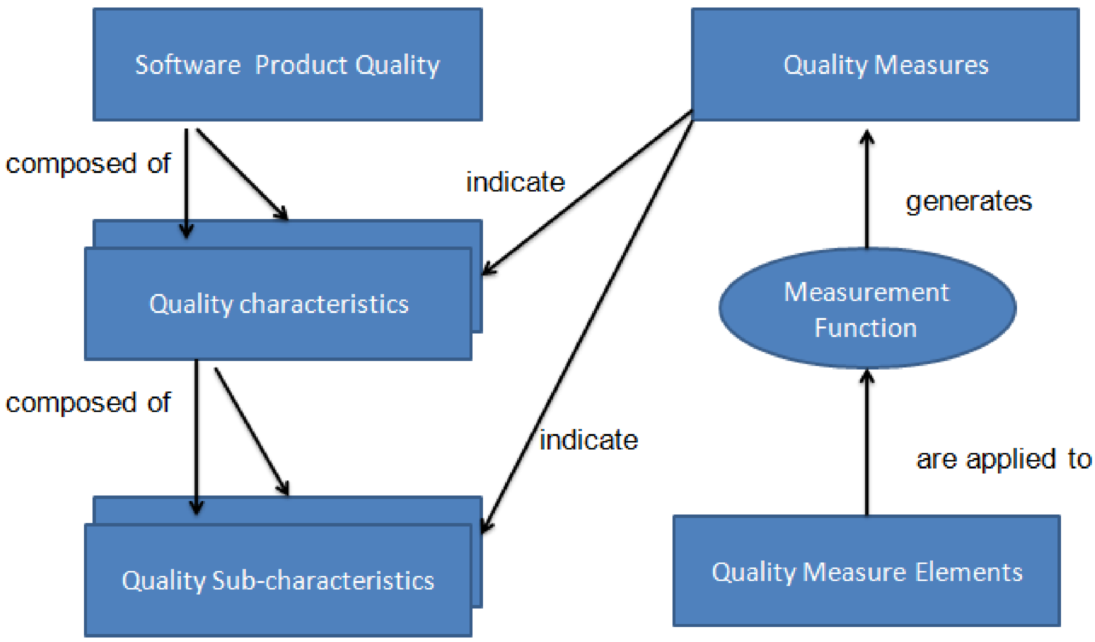


Figure 3-4 Software Product Quality measurement Reference Model (ISO/IEC 25020, 2006)

3.5. Software quality characteristics and sub-characteristics in SQuaRE.

Currently there are three quality models in the SQuaRE series: the quality in use model and the product quality model in this International Standard, and the data quality model in ISO/IEC 25010. The quality models together serve as a framework to ensure that all characteristics of quality are considered. These models provide a set of quality characteristics relevant to a wide range of stakeholders, such as: software developers, system integrators, acquirers, owners, maintainers, contractors, quality assurance and control professionals, and users. [VANÍČEK, 2010]

3.5.1. Quality in use model.

The quality in use model defines five characteristics related to outcomes of interaction with a system: effectiveness, efficiency, satisfaction, freedom from risk, and context coverage (Figure 3-5). Each characteristic can be assigned to different activities of stakeholders, for example, the interaction of an operator or the maintenance of a developer. [VANÍČEK, 2010]

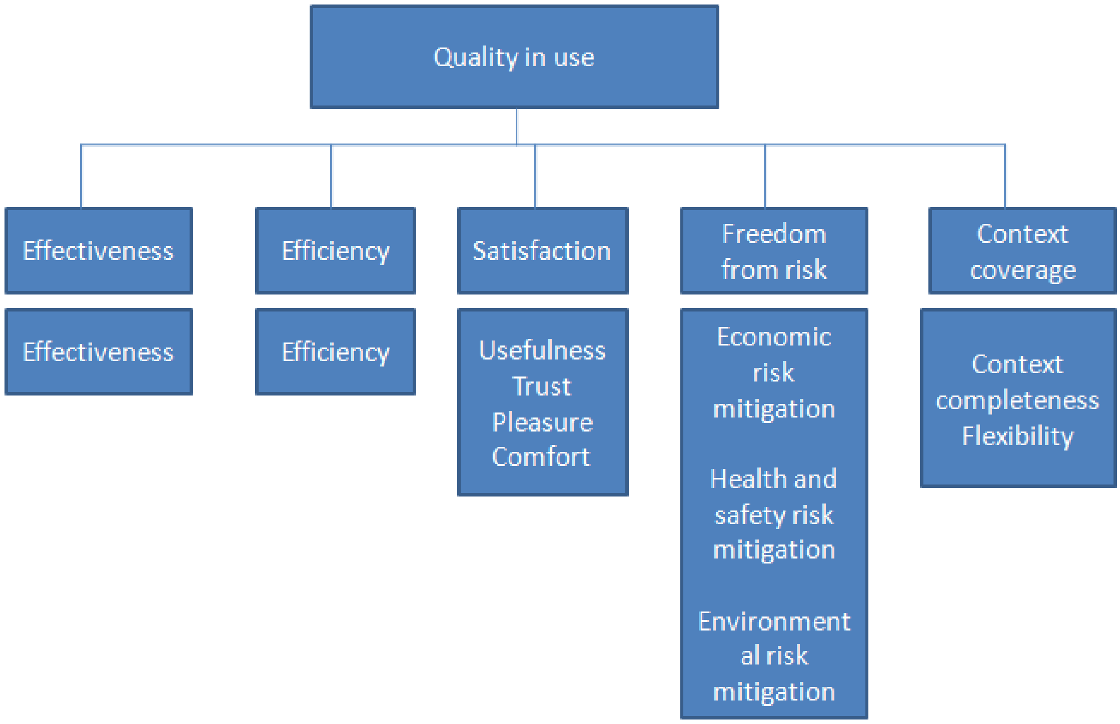


Figure 3-5 Quality in use model (Source: ISO/IEC 25010, 2010)

The quality in use of a system characterizes the impact that the product (system or software product) has on stakeholders. It is determined by the quality of the software, hardware and operating environment, and the characteristics of the users, tasks and social environment. All these factors contribute to the quality in use of the system. [VANÍČEK, 2010]

Quality in use is the degree to which a product or a system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, freedom from risk and satisfaction in specific contexts of use. [ISO, 2010]

Effectiveness: accuracy and completeness with which users achieve specified goals. [ISO 9241 – 11]

Efficiency: resources expended in relation to the accuracy and completeness with which users achieve goal. [ISO 9241 – 11]

NOTE Relevant resources can include time to complete the task (human resources), materials, or the financial cost of usage.

Satisfaction: degree to which user needs are satisfied when a product or system is used in a specified context of use.

NOTE 1 For a user who does not directly interact with the product or system, only purpose accomplishment and trust are relevant.

NOTE 2 Satisfaction is the user’s response to interaction with the product or system, and includes attitudes towards use of the product.

Usefulness: degree to which a user is satisfied with their perceived achievement of pragmatic goals, including the results of use and the consequences of use.

Trust: degree to which a user or other stakeholder has confidence that a product or system will behave as intended.

Pleasure: degree to which a user obtains pleasure from fulfilling their personal needs.

NOTE Personal needs can include needs to acquire new knowledge and skills, to communicate personal identity and to provoke pleasant memories.

Comfort: degree to which the user is satisfied with physical comfort

Freedom from risk: degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment.

NOTE Risk is a function of the probability of occurrence of a given threat and the potential adverse consequences of that threat's occurrence.

Economic risk mitigation: degree to which a product or system mitigates the potential risk to financial status, efficient operation, commercial property, reputation or other resources in the intended contexts of use.

Health and safety risk mitigation: degree to which a product or system mitigates the potential risk to people in the intended contexts of use.

Environmental risk mitigation: degree to which a product or system mitigates the potential risk to property or the environment in the intended contexts of use.

Context coverage: degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified.

NOTE Context of use is relevant to both quality in use and some product quality sub-characteristics (where it is referred to as “specified conditions”).

Context completeness: degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in all the specified contexts of use.

NOTE Context completeness can be specified or measured either as the degree to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, freedom from risk and satisfaction in all the intended contexts of use, or by the presence of product properties that support use in all the intended contexts of use.

EXAMPLE The extent to which software is usable using a small screen, with low network bandwidth, by a non-expert user; and in a fault-tolerant mode (e.g. no network connectivity).

Flexibility: degree to which a product or system can be used with effectiveness, efficiency, freedom from risk and satisfaction in contexts beyond those initially specified in the requirements.

NOTE 1 Flexibility can be achieved by adapting a product for additional user groups, tasks and cultures.

NOTE 2 Flexibility enables products to take account of circumstances, opportunities and individual preferences that might not have been anticipated in advance.

NOTE 3 If a product is not designed for flexibility, it might not be safe to use the product in unintended contexts.

NOTE 4 Flexibility can be measured either as the extent to which a product can be used by additional types of users to achieve additional types of goals with effectiveness, efficiency, freedom from risk and satisfaction in additional types of contexts of use, or by a capability to be modified to support adaptation for new types of users, tasks and environments, and suitability for individualization as defined in ISO 9241-110.

3.5.2. Product Quality Model.

The product quality model categorizes system/software product quality properties into eight characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability. Each characteristic is composed of a set of related sub-characteristics (Figure 3-6).

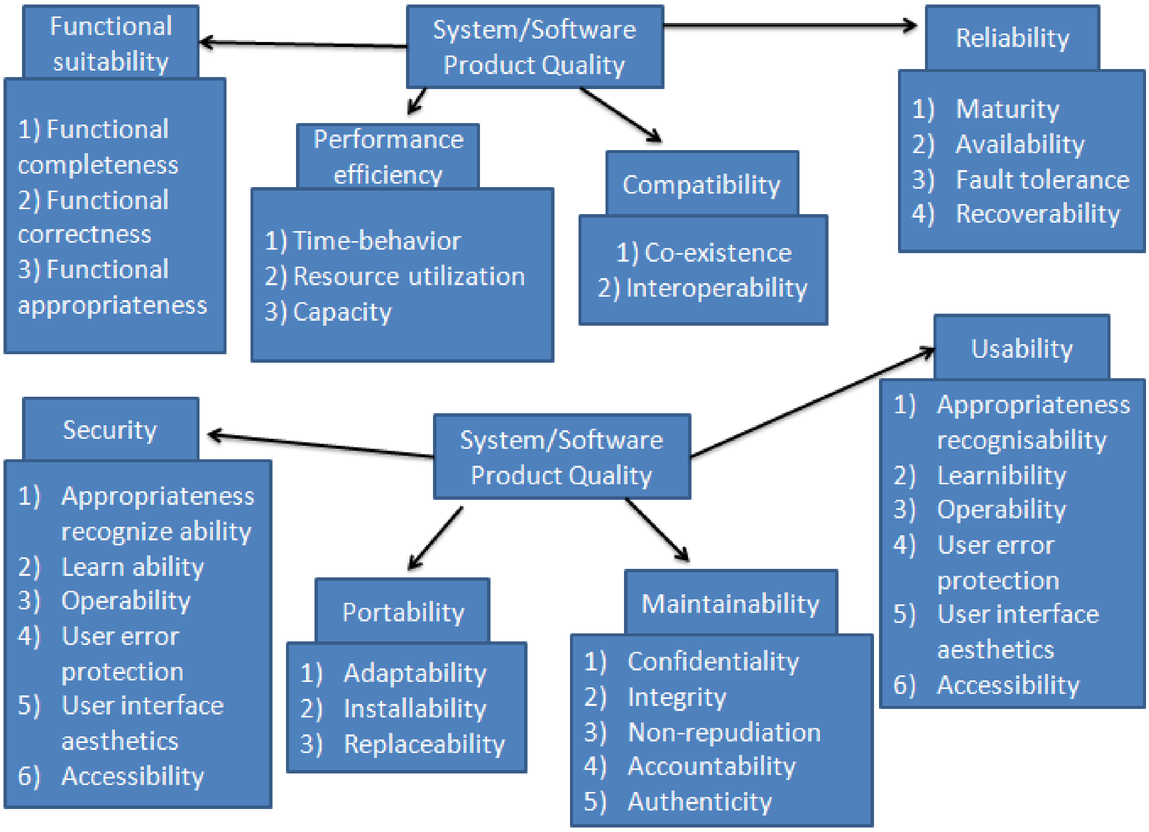


Figure 3-6 Product Quality characteristics and sub-characteristics (Source: ISO/IEC 25010, 2010)

The product quality model can be applied to just a software product, or to a computer system that includes software, as most of the sub-characteristics are relevant to both software and systems. [ISO, 2010]

Functional suitability: degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.

NOTE Functional suitability is only concerned with whether the functions meet stated and implied needs, not the functional specification.

Functional completeness: degree to which the set of functions covers all the specified tasks and user objectives.

Functional correctness: degree to which a product or system provides the correct results with the needed degree of precision.

Functional appropriateness: degree to which the functions facilitate the accomplishment of specified tasks and objectives.

EXAMPLE A user is only presented with the necessary steps to complete a task, excluding any unnecessary steps.

NOTE Functional appropriateness corresponds to suitability for the task in ISO 9241-110.

Performance efficiency: performance relative to the amount of resources used under stated conditions.

NOTE Resources can include other software products, the software and hardware configuration of the system, and materials (e.g. print paper, storage media).

Time behavior: degree to which the response and processing times and throughput rates of a product or system, when performing its functions, meet requirements.

Resource utilization: degree to which the amounts and types of resources used by a product or system when performing its functions meet requirements.

NOTE Human resources are included as part of **efficiency.**

Capacity: degree to which the maximum limits of a product or system parameter meet requirements.

NOTE Parameters can include the number of items that can be stored, the number of concurrent users, the communication bandwidth, throughput of transactions, and size of database.

Compatibility: degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.

NOTE Adapted from ISO/IEC/IEEE 24765.

Co-existence: degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product.

Interoperability: degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.

NOTE Based on ISO/IEC/IEEE 24765.

Usability: degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use

NOTE 1 Adapted from ISO 9241-210.

NOTE 2 Usability can either be specified or measured as a product quality characteristic in terms of its sub-characteristics, or specified or measured directly by measures that are a subset of quality in use.

Appropriateness recognizability: degree to which users can recognize whether a product or system is appropriate for their needs.

Cf. **functional appropriateness**.

NOTE 1 Appropriateness recognizability will depend on the ability to recognize the appropriateness of the product or system's functions from initial impressions of the product or system and/or any associated documentation.

NOTE 2 The information provided by the product or system can include demonstrations, tutorials, documentation or, for a web site, the information on the home page.

Learnability: degree to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use.

NOTE Can be specified or measured either as the extent to which a product or system can be used by specified users to achieve specified goals of learning to use the product or system with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use, or by product properties corresponding to suitability for learning as defined in ISO 9241-110.

Operability: degree to which a product or system has attributes that make it easy to operate and control NOTE Operability corresponds to controllability, (operator) error tolerance and conformity with user expectations as defined in ISO 9241-110.

User error protection: degree to which a system protects users against making errors.

User interface aesthetics: degree to which a user interface enables pleasing and satisfying interaction for the user.

NOTE This refers to properties of the product or system that increase the pleasure and satisfaction of the user, such as the use of colour and the nature of the graphical design.

Accessibility: degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use.

NOTE 1 The range of capabilities includes disabilities associated with age.

NOTE 2 Accessibility for people with disabilities can be specified or measured either as the extent to which a product or system can be used by users with specified disabilities to achieve specified goals with effectiveness, efficiency, freedom from risk and satisfaction in a specified context of use, or by the presence of product properties that support accessibility.

Reliability: degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

NOTE 1 Adapted from ISO/IEC/IEEE 24765.

NOTE 2 Wear does not occur in software. Limitations in reliability are due to faults in requirements, design and implementation, or due to contextual changes.

NOTE 3 Dependability characteristics include availability and its inherent or external influencing factors, such as availability, reliability (including fault tolerance and recoverability), security (including confidentiality and integrity), maintainability, durability, and maintenance support.

Maturity: degree to which a system meets needs for reliability under normal operation.

NOTE The concept of maturity can also be applied to other quality characteristics to indicate the degree to which they meet required needs under normal operation.

Availability: degree to which a system, product or component is operational and accessible when required for use. [ISO/IEC/IEEE 24765]

NOTE Externally, availability can be assessed by the proportion of total time during which the system, product or component is in an up state. Availability is therefore a combination of maturity (which governs the frequency of failure), fault tolerance and recoverability (which governs the length of down time following each failure).

Fault tolerance: degree to which a system, product or component operates as intended despite the presence of hardware or software faults.

NOTE Adapted from ISO/IEC/IEEE 24765.

Recoverability: degree to which, in the event of an interruption or a failure, a product or system can recover the data directly affected and re-establish the desired state of the system.

NOTE Following a failure, a computer system will sometimes be down for a period of time, the length of which is determined by its recoverability.

Security: degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.

NOTE 1 As well as data stored in or by a product or system, security also applies to data in transmission.

NOTE 2 Survivability (the degree to which a product or system continues to fulfill its mission by providing essential services in a timely manner in spite of the presence of attacks) is covered by **recoverability**.

NOTE 3 Immunity (the degree to which a product or system is resistant to attack) is covered by **integrity**.

NOTE 4 Security contributes to **trust**.

Confidentiality: degree to which a product or system ensures that data are accessible only to those authorized to have access.

Integrity: degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data. [ISO/IEC/IEEE 24765]

Non-repudiation: degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later.

NOTE Adapted from ISO 7498-2:1989.

Accountability: degree to which the actions of an entity can be traced uniquely to the entity.

NOTE Adapted from ISO 7498-2:1989.

Authenticity: degree to which the identity of a subject or resource can be proved to be the one claimed.

NOTE Adapted from ISO/IEC 13335-1:2004.

Maintainability: degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

NOTE 1 Modifications can include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications. Modifications include those carried out by specialized support staff, and those carried out by business or operational staff, or end users.

NOTE 2 Maintainability includes installation of updates and upgrades.

NOTE 3 Maintainability can be interpreted as either an inherent capability of the product or system to facilitate maintenance activities, or the quality in use experienced by the maintainers for the goal of maintaining the product or system.

Modularity: degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.

[ISO/IEC/IEEE 24765]

Reusability: degree to which an asset can be used in more than one system, or in building other assets.

NOTE Adapted from IEEE 1517-2004.

Analyzability: degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified.

NOTE Implementation can include providing mechanisms for the product or system to analyze its own faults and provide reports prior to a failure or other event.

Modifiability: degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.

NOTE 1 Implementation includes coding, designing, documenting and verifying changes.

NOTE 2 **Modularity** and **analyzability** can influence modifiability.

NOTE 3 Modifiability is a combination of changeability and stability.

Testability: degree of effectiveness and efficiency with which test criteria can be established for a system, product or component and tests can be performed to determine whether those criteria have been met.

NOTE Adapted from ISO/IEC/IEEE 24765.

Portability: degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

NOTE 1 Adapted from ISO/IEC/IEEE 24765.

NOTE 2 Portability can be interpreted as either an inherent capability of the product or system to facilitate porting activities, or the quality in use experienced for the goal of porting the product or system.

Adaptability: degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environments.

NOTE 1 Adaptability includes the scalability of internal capacity (e.g. screen fields, tables, transaction volumes, report formats, etc.).

NOTE 2 Adaptations include those carried out by specialized support staff, and those carried out by business or operational staff, or end users.

NOTE 3 If the system is to be adapted by the end user, adaptability corresponds to suitability for individualization as defined in ISO 9241-110.

Installability: degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment.

NOTE If the product or system is to be installed by an end user, installability can affect the resulting functional appropriateness and operability.

Replaceability: degree to which a product can be replaced by another specified software product for the same purpose in the same environment.

EXAMPLE The replaceability of a new version of a software product is important to the user when upgrading.

NOTE 1 Replaceability can include attributes of both installability and adaptability. The concept has been introduced as a sub-characteristic of its own because of its importance.

NOTE 2 Replaceability will reduce lock-in risk: so that other software products can be used in place of the present one, for example by the use of standardized file formats.

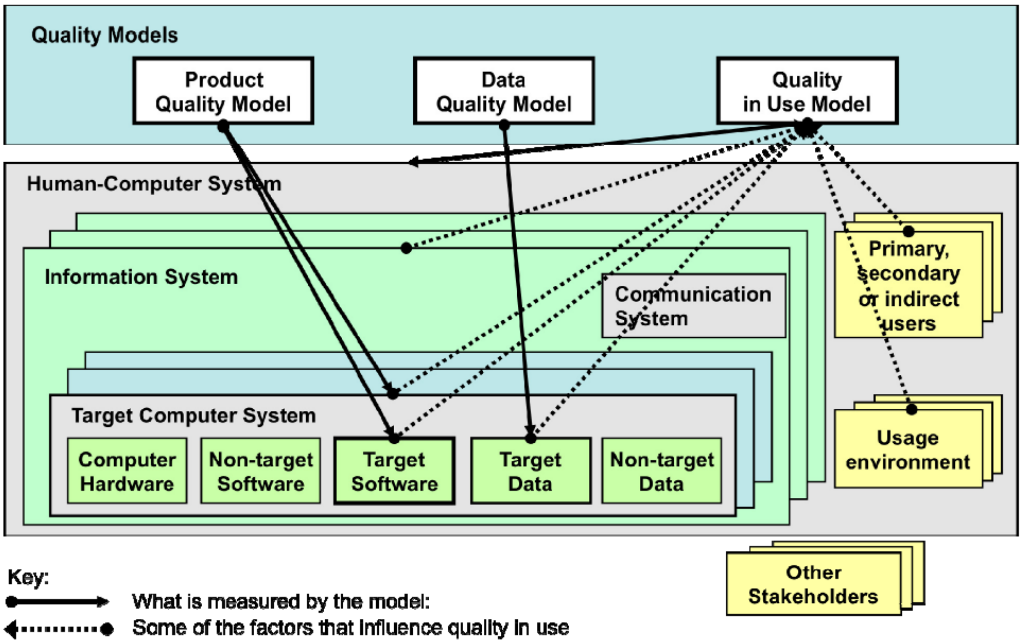


Figure 3-7 Targets of the quality models (Source: ISO/IEC 25010, 2010)

Figure 3-7 illustrates the targets of the quality models and the related entities. The product quality model focuses on the target computer system that includes the target software product and the quality in use model focuses on the whole human-computer system that includes the target computer system and target software product. The target computer system also includes computer hardware, non-target software products, non-target data, and target data, which is the subject of the data quality model. [ISO, 2010]

The target computer system is included in an information system that can also include one or more computer systems and communication systems, such as a local area network and the Internet. The information system is within a wider human-computer system (such as an enterprise system, embedded system or large-scale control system) and can include users and the technical and physical usage environment. Where the boundary of the system is judged to be, depends upon the scope of the requirements or evaluation, and upon who the users are. [ISO, 2010]

3.5.3. Using a quality model

The product quality and quality in use models are useful for specifying requirements, establishing measures, and performing quality evaluations. The defined quality characteristics can be used as a checklist for ensuring a comprehensive treatment of quality requirements, thus providing a basis for estimating the consequent effort and activities that will be needed during systems development. The characteristics in the quality in use model and product quality model are intended to be used as a set when specifying or evaluating computer system or software product quality. [VANÍČEK, 2010]

It is not practically possible to specify or measure all sub-characteristics for all parts of a large computer system or software product. Similarly it is not usually practical to specify or measure quality in use for all possible user-task scenarios. The relative importance of quality characteristics will depend on the high-level goals and objectives for the project. Therefore the model should be tailored before use as part of the decomposition of requirements to identify those characteristics and sub-characteristics that are most important, and resources allocated between the different types of measure depending on the stakeholder goals and objectives for the product. [VANÍČEK, 2010]

3.5.4. Relationship between the models.

The properties of the software product and computer system determine the product quality in particular context of use. (Table 3-1) [ISO, 2010]

The functional suitability, performance efficiency, usability, reliability and security we have a significant influence on the quality in use for primary users. Performance efficiency, reliability and security can also be specific concerns of other stakeholders who specialize in these areas. Compatibility, maintainability and portability will have a significant influence on quality in use for secondary users who maintain the systems. [ISO, 2010]

Table 3-1 Influence of quality characteristics (Source: ISO/IEC 25010, 2010)

Software product properties	Computer system properties	Product Quality characteristic	Influence on quality in use for primary users	Influence on quality in use for maintenance tasks	Information system quality concerns of other stakeholders
→	→	Functional suitability	*		
→	→	Performance efficiency	*		*
→	→	Compatibility		*	
→	→	Usability	*		
→	→	Reliability	*		*
→	→	Security	*		*
→	→	Maintainability		*	
→	→	Portability		*	

Key: →These properties influence product quality

* Product quality influences quality in use for these stakeholders

Alongside with the ISO 2510 [2010] Quality model there are exist several quality models such as:

1) McCall's Quality Model
2) Dromey's Quality Model
3) Boehm's Quality Model
4) FURPS Quality Model
5) ISO 9126 Quality Model

All this models are used to measure quality of software systems and software products.

3.5.5. McCall's Quality Model

McCall's Quality model (also known as General Electric model of 1977) is one of the most known quality models in the software engineering literature. It has been presented by Jim McCall in 1977 et al [1977]. This model is originates from the US military and is primarily aimed towards the system developers and the system development process. Using this model, McCall attempts to bridge the gap between users and developers by focusing on a number of software quality factors that reflect both the users' and the developers' priorities. [McCALL et al, 1977]

The structure of the McCall's quality model consists of three major perspectives (types of quality characteristics) for defining and identifying the quality of a software product, and each of these major perspectives consists of a number of quality factors. Each of these quality factors has a step of quality criteria, and each quality criteria could be reflected by one or more measures, see Figure 3-8 for the details of the McCall's quality model structure. [McCALL et al, 1977] The contents of three major perspectives are the following:

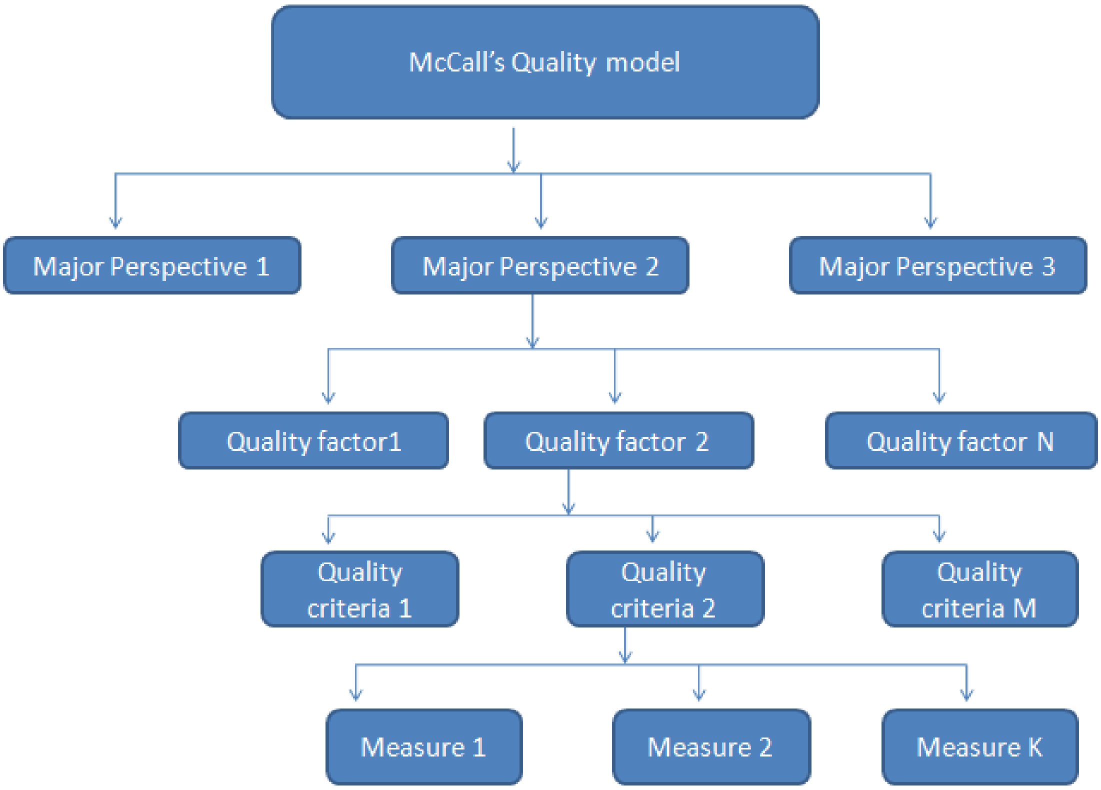


Figure 3-8 Structure of McCall's Quality Model (Source: McCall, 1997)

Product Revision: it is about identification quality factors that would be influence the ability to change the software product, and it includes:

1) Maintainability: the effort required to detect, locate and fix a fault in the program within its operating environment
2) Flexibility: the ease of making changes required by changes in the operating environment
3) Testability: the ease of testing the program, to ensure that it is error-free and meets its specification.

Product Operations: it is about the characteristics of the product operation. The quality of the product operations depends on:

1) Correctness: the extent to which a program fulfils its specification
2) Reliability: the system ability not to fail
3) Efficiency: it further categorized into execution efficiency and storage efficiency and generally meaning the use of resources, e.g. processor time, storage.

4) Integrity: the protection of the program from unauthorized access.
5) Usability: the ease of the use of the software.

Product Transition: it is about the adaptability of the product to new environments. It is all about:

1) Portability: the effort required to transfer a program from one environment to another.
2) Reusability: the ease of reusing software in a different context.
3) Interoperability: the effort required to couple the system to another system.

In more details, McCall's Quality Model consists of 11 quality factors to describe the external view of the software (from user's point of view), 23 quality criteria to describe the internal view of the software (from developer's view) and a set of metrics which are defined and used to provide a scale and method for measurement. Table 3-2 presents two of 3 major perspectives and their corresponding quality factors and quality criteria.

The main objective of the McCall's quality model is that the quality factors structure should provide a complete software quality picture. [Kitchenham, 1996]

The actual quality metric is computed by answering "yes" and "no" questions.

However, if answering equally amount of "yes" and "no" on the questions measuring a quality criteria, then you will achieve 50% on that quality criteria. [RAFA, 2010]

Table 3-2 Concepts of McCall's Quality Model – Major perspectives, quality factors and quality criteria

Major Perspectives	Quality factors	Quality criteria
Product revision	Maintainability	Simplicity Conciseness Self-descriptiveness Modularity

	Flexibility	Self-descriptiveness Expandability Generality
	Testability	Simplicity Instrumentation Self-descriptiveness Modularity
Product operations	Correctness	Traceability Completeness Consistency
	Efficiency	Execution efficiency Storage efficiency
	Reliability	Consistency Accuracy Error tolerance
	Integrity	Access control Access audit
	Usability	Operability Training Communicativeness

3.5.6. Boehm's Quality Model

Boehm [1976, 1978] introduced his quality to automatically and quantitatively evaluate the quality of software. This model attempts to qualitatively define the quality of software by a predefined set of attributes and metrics. It consists of high-level characteristics, intermediate-level characteristics and lowest-level (primitive) characteristics which contribute to the overall quality level (see Figure 3-9).

In this model, the high-level requirements of actual use to which evaluation of software quality could be put. In its high-level, there are three characteristics, that is [BOEHM et al, 1976, BOEHM et al,1978]:

1. As – is utility: to address how well, easily, reliably and efficiently can I use the software product as – is?
2. Maintainability: to address how easy is it to understand, modify and retest the software product?
3. Portability: to address if can I still use the software product when the environment has been changed?

Table 3-3 shows the contents of the Boehm's quality model in the tree levels, high-level, intermediate-level and lowest-level characteristics. In addition, it is noted, that there is a number of the lowest-level characteristics which can be related to more than one intermediate-level characteristics, for example, the 'Self Contentedness' primitive characteristic could be related to the 'reliability' and 'portability' primitive characteristics.

In the intermediate level characteristic, there are seven quality characteristics that together represent the qualities expected from a software system [BOEHM et al, 1976, BOEHM et al,1978]:

1) Portability: the software can be operated easily and well on computer configurations other than its current one.
2) Reliability: the software can be expected to perform its intended functions satisfactory.
3) Efficiency: the software fulfills its purpose without waste of resources.
4) Usability: the software is reliable, efficient and human-engineered.

5) Testability: the software is facilitates the establishment of verification criteria and supports evaluation of its performance.
6) Understandability: the software purpose is clear to the inspector.
7) Flexibility: the software facilitates the incorporation of changes, once the nature of the desired change has been determined.

The primitive characteristics can be used to provide the foundation for defining quality metrics, this use is one of the most important goals established by Boehm when he constructed his quality model. [RAFA, 2010]

One or more metrics are supposed to measure a given primitive characteristic. Boehm [1978] defined the 'metric' as "a measure of extent or degree to which a product possesses and exhibits a certain (quality) characteristic".

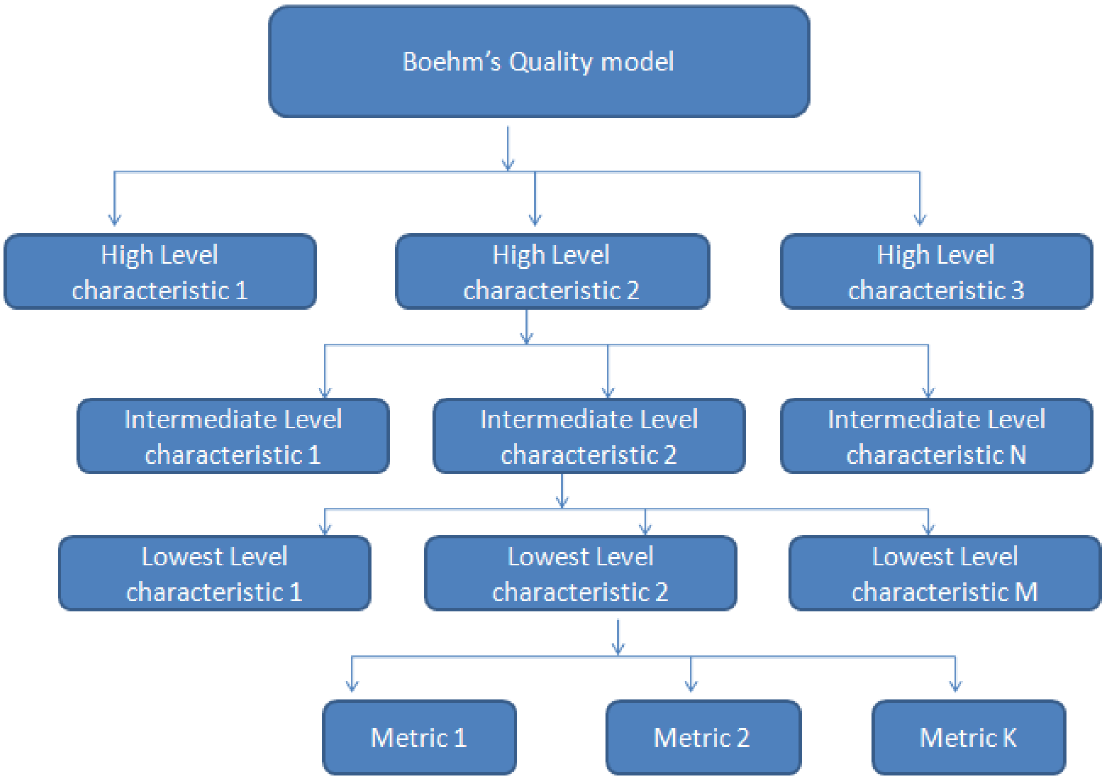


Figure 3-9 Structure of Boehm's Quality Model (Source: Boehm, 1978)

Table 3-3 The contents of Boehm's Quality model (Source: Boehm et. Al, 1976, Boehm et al, 1978)

High- level characteristics (3)	Intermediate–level characteristics (7)	Primitive Characteristics (15)
As-is Utility	Reliability	Self Containtedness Accuracy Completeness Robustness/Integrity Consistency
	Efficiency	Accountability Device efficiency Accessibility
	Human Engineering	Robustness/Integrity Accessibility
Portability		Device independence Self Containtedness
Maintainability	Testability	Accountability Communicativeness Self Descriptiveness Structuredness
	Understandability	Consistency Structuredness Conciseness Legibility

	Modifiability	Structuredness Augmentability

3.5.7. Dromey's Quality Model

This quality model has been presented by Dromey [1995, 1996]. It is a product based quality model that recognizes that quality evaluation differs for each product and that a more dynamic idea for modeling the process is needed to be wide enough to apply for different systems [DROMEY, 1995].

Furthermore, Figure 3-10 shows that it consists of four software product properties and for each property there is a number of quality attributes. In addition, Figure 3-11 shows the contents of the Dromey's quality model.

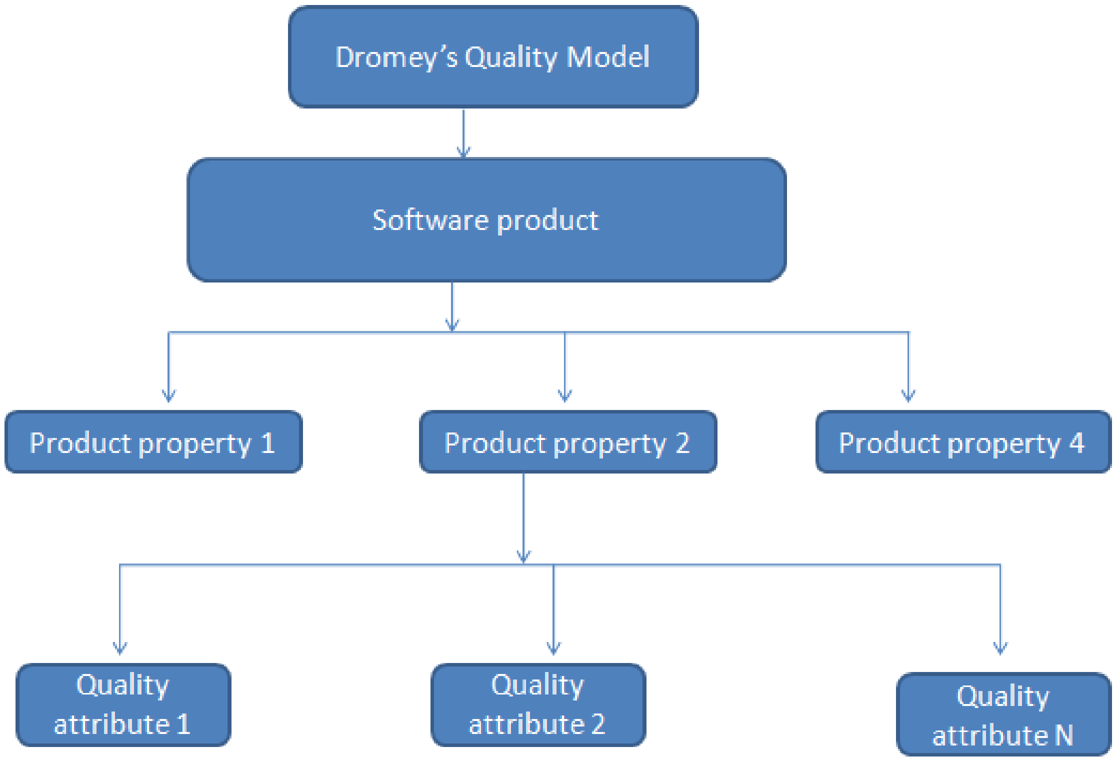


Figure 3-10 Structure of Dromey's Quality Model (Source: Dromey, 1995)

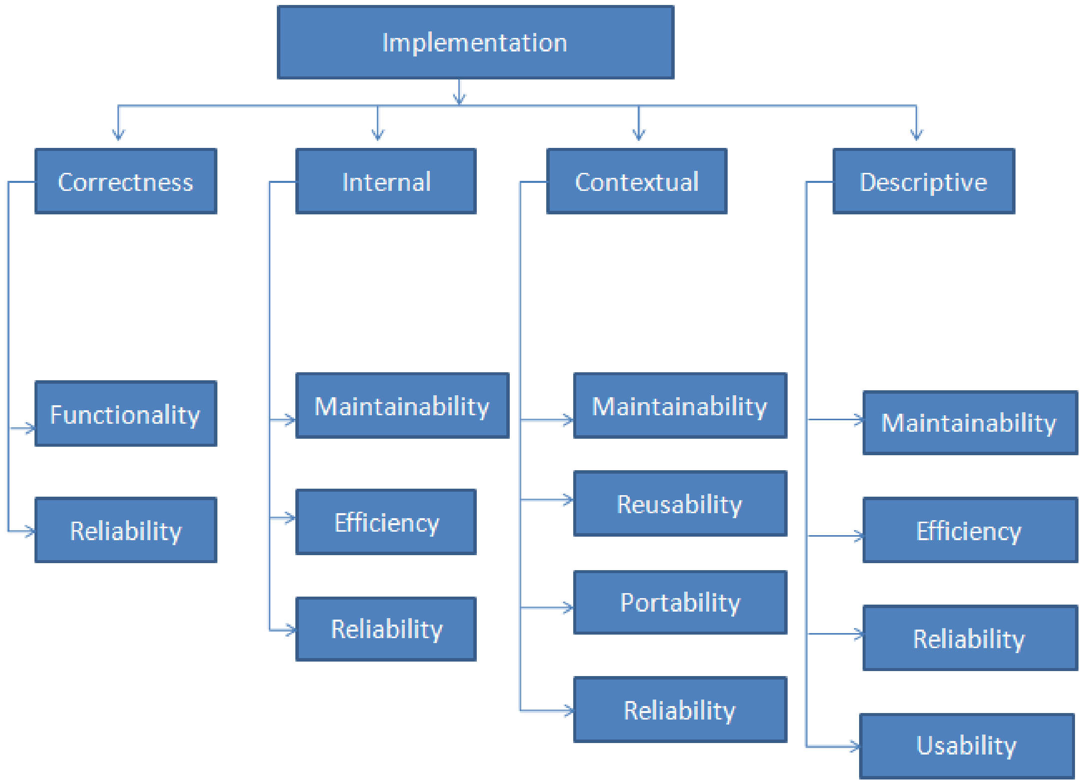


Figure 3-11 Characteristics and sub-characteristics of Dromey's Quality Model (Source: Dromey, 1995)

3.5.8. FURPS Quality model

The FURPS model originally presented by Robert Grady [1992], then it has been extended by IBM Rational Software [JACOBSON et al, 1999, KRUCHTEN, 2000] into FURPS+, where the'+' indicates such requirements as design constraints, implementation requirements, interface requirements and physical requirements [JACOBSON et al, 1999].

In this quality model, the FURPS stands for [GRADY, 1992] –as in Figure 3-12 – the following five characteristics:

1) Functionality: it may include feature sets, capabilities, and security.
2) Usability: it may include human factors, aesthetics, consistency in the user interface, online and context sensitive help, wizards and agents, user documentation, and training materials.
3) Reliability: it may include frequency and severity of failure, recoverability, predictability, accuracy. And mean time between failures (MTBF).

4) Performance: it imposes conditions on functional requirements such as speed, efficiency, availability, accuracy, throughput, response time, recovery time, and resource usage.
5) Supportability: it may include testability, extensibility, adaptability, maintainability, compatibility, configurability, serviceability, installability, and localizability.

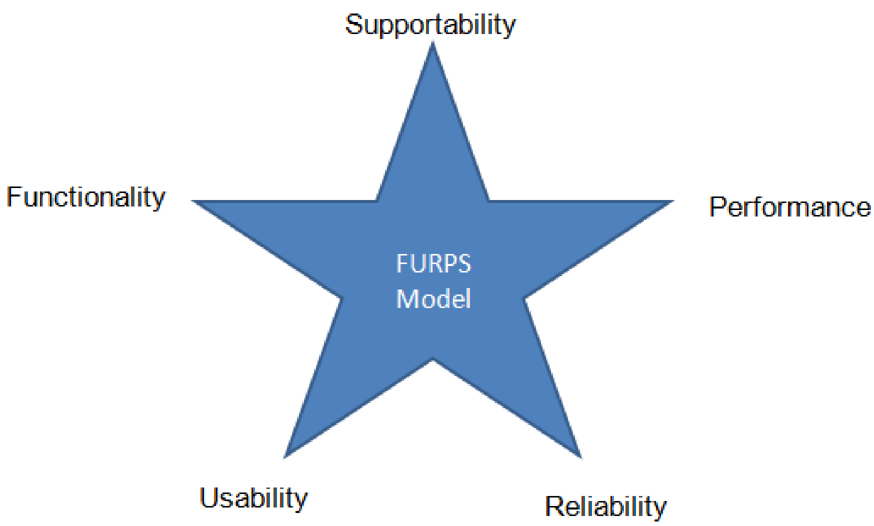


Figure 3-12 Structure of FURPS Quality Model (Source: Grady, 1992)

3.5.9. ISO 9126 Quality model.

In 1991, the ISO published its first international consensus on the terminology for the quality characteristics for software product evaluation; this standard was called as Software Product Evaluation – Quality Characteristics and Guidelines for their use (ISO 9126).

From 2001 to 2004, the ISO published an expanded version, containing both ISO quality models and inventories of proposed measures for these models.

The first document of the ISO 9126 series – Quality Model – contains two-parts quality model for software product quality [ISO, 2001]

1. Internal and external quality model.
2. Quality in use model.

The first part of the two-part quality model determines six characteristics in which they are subdivided into twenty-seven sub-characteristics for internal and external quality, as in Figure 3-13 [ISO, 2001]. These sub-characteristics are a result of internal software attributes and are noticeable externally when the software is used as a part of a computer system. The second part of the two-part model indicates for quality in use characteristics, as in Figure 3-14 [ISO, 2001].

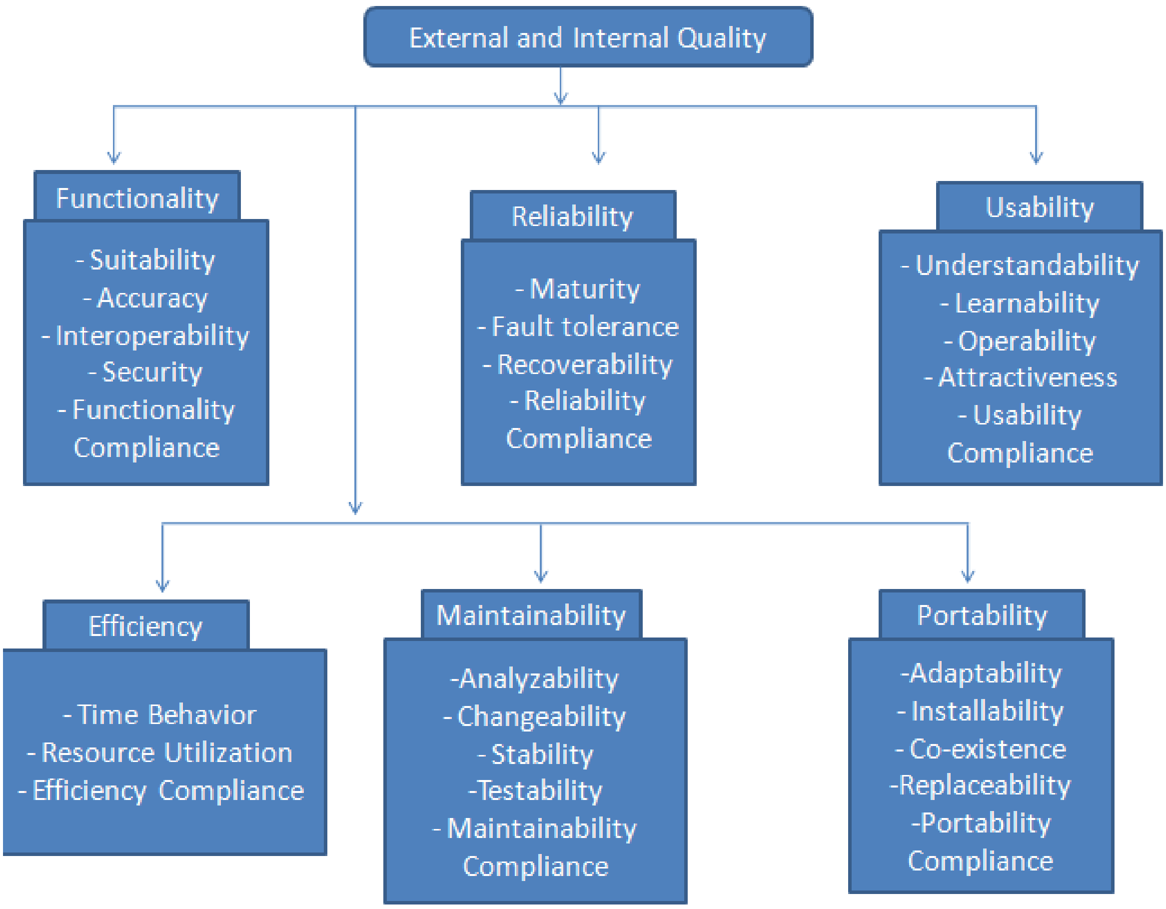


Figure 3-13 ISO 9126 Quality model structure for external and internal quality (Source ISO/IEC 9126, 2001)

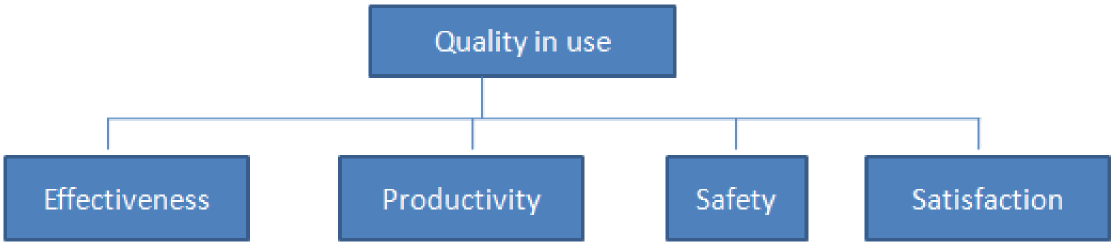


Figure 3-14 ISO 9126 quality model for quality in use (characteristics) (Source: ISO/IEC 9126, 2001)

Figure 3-15 shows the ISO view of the expected relationship between internal, external, and quality in use attributes. The internal quality attributes while the external attributes influences on the quality in use attributes. Furthermore, the quality in use depends on the external quality while the external quality depends on the international quality [ISO, 2010].

For the internal and external software products, each quality characteristics and its corresponding sub-characteristics are defined in ISO 25010 [ISO/IEC, 2010] as follows:

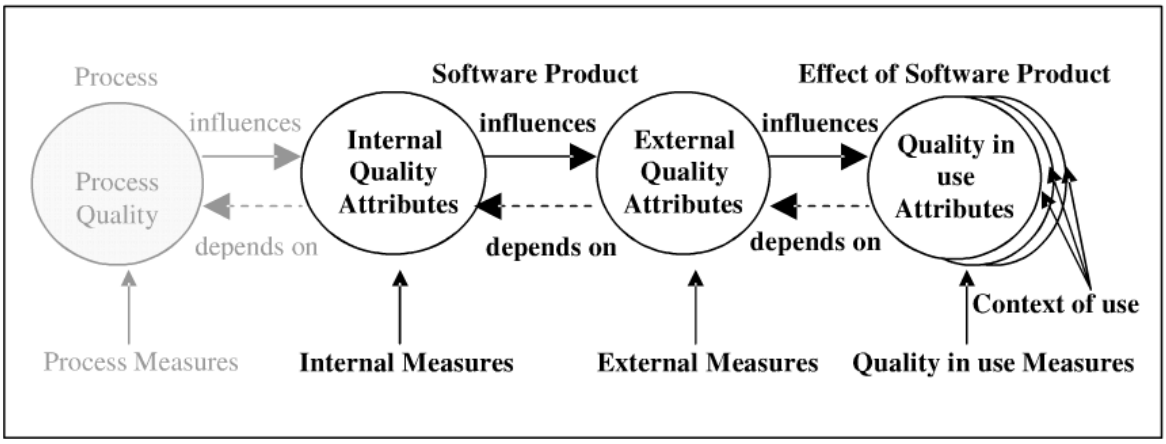


Figure 3-15 Quality in the lifecycle (Source: ISO, 2001)

3.6. Analysis of the Quality models

In this section, a comparison between the availability of the characteristics (called factors or attributes in some quality models) within the six quality models will be presented. Table 4 presented this comparison, at the end this table you will find the number of the corresponding characteristics for each quality model.

From the 18 characteristics, only one is common to all quality models that is, the ‘reliability’. Also, there are only two characteristics (i.e. ‘efficiency’, ‘usability’) which are belonging to five quality models. Three characteristics are common to four quality models that is, the ‘functionality’, ‘portability’ and ‘maintainability’ characteristics. Two characteristics belong only to two quality models, that is, the ‘testability’ and ‘performance’ characteristics. And, nine characteristics (i.e. ’flexibility’, ‘correctness’, ‘integrity’ in McCall’s quality model; ‘human engineering’, ‘understandability’ and ‘modifiability’ in Boehm’s quality model; ‘supportability’ in FURPS quality model; ‘security’ and ‘compatibility’ ISO 25010) are defined in only one quality model.

Furthermore, it can be noted that the ‘testability’ and ‘understandability’ are used as factors/attributes/characteristics in some quality models. However, in ISO 9126-1 and ISO 25010, these factors/attributes/characteristics are defined as sub-characteristics. More specifically, the ‘testability’ is belonging to the ‘maintainability’ characteristic. The ‘understandability’ is belonging to the ‘usability’ characteristic.

From my point of view, the ISO 25010 quality model is the most useful one since it had been built based on an international consensus and agreement from all the country member of the ISO organization and it is replacement of ISO 9126-1.

Table 3-4 Comparisons between the six quality models

Factors/Attributes/Characteristics	McCall	Boehm	Dromey	FURPS	ISO 9126	ISO 25010
1. Maintainability	*		*		*	*
2. Flexibility	*					
3. Testability	*	*				
4. Correctness	*					
5. Efficiency	*	*	*		*	*
6. Reliability	*	*	*	*	*	*
7. Integrity	*					
8. Usability	*		*	*	*	*
9. Portability		*	*		*	*
10. Reusability			*			
11. Human Engineering		*				
12. Understandability		*				
13. Modifiability		*				
14. Functionality			*	*	*	*
15. Performance				*		*
16. Supportability				*		
17. Compatibility						*
18. Security						*
18	8	7	7	5	6	9

There are a number of quality models in software engineering literature, each one of these quality models consists of a number of quality characteristics (or factors, as called in some models). These quality characteristics could be used to reflect the quality of the software product from the view of that characteristic. Selecting which one of the quality models use is a real challenge. [RAFA, 2010]

1. McCall's Quality Model
2. Boehm's Quality Model

3. Dromey's Quality Model
4. FURPS Quality Model
5. ISO 9126 Quality Model
6. ISO 25010 Quality Model

From comparing of these 6 Quality models we can conclude the following comments:

1. In McCall's quality model, the quality is subjectively measured based on the judgment on the person(s) answering the questions ('yes' or 'no' questions). [RAFA, 2010]

2. Three of the characteristics are used in the ISO 9126-1 and ISO 25010 quality models as sub-characteristics from other characteristics.

3. The FURPS quality model is built and extended to be used in the IBM Rational Software Company. Therefore, it is a special – purpose quality model, that is, for the benefits of that company. [RAFA, 2010]

4. The metrics in the lower level of the McCall's, Boehm's, Dromey's and FURPS quality models are neither clearly nor completely defined and connected to the upper level of the quality models. For example, in McCall's quality model, the measures should be clearly defined and connected to the corresponding quality criteria, see Figure 3-8. [RAFA, 2010]

The ISO 25010 quality model is the most useful and the new one since it has been based on an international consensus and agreement from all the country members of the ISO organization.

4. The Case study: comparison of the two software products for accounting used in Russia and in the Czech Republic.

This chapter is focused on practical part how to use ISO standard in reality. First of all were measured quality characteristics and sub-characteristics for both software products. However on practice there are a lot of problems which are connected with difficulties to measure quality characteristics. The main problem is that evaluation criteria for all characteristics and sub-characteristics are not established. There are no free tools in the internet which will help to measure all characteristics of quality according to ISO standard. Different companies are offering their services for quality evaluation of software product. But regularly these services are quite expensive.

For example, price for the evaluation software report from Technology Evaluation Centers started from 675$ and higher.

Specialist can test software using other software programs such as AutomatedQA TestComplete which can test functional abilities of software and run several tests at the same time.

The price for this software program is following:

AutomatedAQ TestComplete 8 version

Standard edition: node-locked license – 999$, floating license – 2999$;

Enterprise edition: node-locked license – 1999$ and for floating license – 4499$.

Rational ClearCase (IBM software product)

An industry-leading solution that provides sophisticated version control, workspace management, parallel development support and build auditing to improve productivity.

User license – 4880$

Specialist who can evaluate software characteristics are in good demand nowadays.

Evaluator can analyze and compare software products between each other after measurement of software quality.

Each of characteristics in quality model describes the specified part of quality and contains several sub-characteristics. Each of sub-characteristics should answers for a specified questions and measures part of main characteristic.

Evaluation process will be given on example of two accounting software products.

First of all we should know the definition of accountancy and accounting software:

Accountancy is the process of communicating financial information about business entity to users such as shareholder and managers. The communication generally in the form of financial statements that show in money terms the economic resources under the control of management; the art lies in selecting the information that is relevant to the user and is reliable. Accountancy is a branch of mathematical science that is useful in discovering the causes of success and failure in the business. The principles of accountancy are applied to business entities in three divisions of practical art, named accounting, bookkeeping and auditing. [Wikipedia]

Accounting software is application software that records and process accounting transaction within functional modules such as accounts payable, accounts receivable, payroll, and trial balance. It functions as an accounting information system. It may be developed in-house by the company or organization using it, may be purchased from a third party, or maybe a combination of a third party application software package with local modifications. It varies greatly in its complexity and cost. [Wikipedia]

4.1. Overview of "1C Accounting" software product

"1C Accounting" is a software product of "1C" Company which has been established in 1991. The company operates in the area of development software products, publishing studying materials, support of mass –market software.

"1C Accounting": a part of "1C: Enterprise" the system of software programs which are covered various business tasks such as CRM, HRM, business and management accounting, etc. It consists of two main parts: framework and specific configurations. The framework itself is not software but is a part of any business application.

The main purpose of "1C Accounting" is automation and generation reports of tax accounting and business accounting according to Russian legislation and IFRS (International Financial Reports Standards).

All reports could be generated according to the Russian legislation, IRFS and according to the business needs as well.

Integral programming language which is a part of platform “1C Accounting 8” allows developers and programmers to implement new algorithms. These algorithms realized new functions. Integral programming language in “1C Accounting 8” is similar to Pascal, Java Script, Basic, but it is not prototype of its.

“1C Accounting 8” tracking Documentation includes in two languages: Russian and English

Description of integral programming language does not include in documentation, because full description of integral programming language includes in help menu in electronic form on both languages: Russian and English.

It is possible to install following language settings:

Russian, English, Bulgarian, Vietnamese, Georgian, Kazakh, Lettish, Lithuanian, German, Romanian, Ukrainian.

There are three versions of “1C Accounting 8”: Basic, Professional and Corporative version. The first basic version contains:

- Applied solution for automation tax and business accounting.
- Business and tax accounting for companies and individual business owners realized in separate databases
- Common system of taxation, single tax on imputed income and applicable simplified taxation.
- Analysis tools: turnover balance sheet, review of accounts, accounting forms, charts, etc.

It can be run only on 1 computer.

It is recommended for small organizations and individual business owners which are used common and applicable simplified taxation.

“1C Accounting 8” Professional version.

The main difference between professional and basic version is that professional version allow maintaining accountancy for several organizations in one database and it is multiuser system.

“1C Accounting 8” Corporate version.

This version has all features from the previous two and includes additional services.

Table 4-1 Configurations of “1C Accounting 8” (Source: “1C” Company)

“1C Accounting 8”	Basic version	Professional version	Corporate version
Applied solution for maintain business and tax accounting	*	*	*
Report preparation for fiscal and tax accounting	*	*	*
Tax accounting including income tax	*	*	*
Tax accounting: common and applicable simple taxation	*	*	*
Ability to customize batch accounts of inventories and contactors payments	*	*	*
Accountability for several organizations in separate databases	*	*	*
Accountability for several organization in common database	_	*	*
Accountability in separate subdivisions	_	_	*
Modifiability of application	_	*	*
Multiuser application	_	*	*
Ability to distribute information databases according to geographical position	_	*	*
Support of COM-connection and Automation-server	_	*	*

Individual business owner or small company can buy Basic version and then upgrade it to Professional or Corporate for additional amount of money. All data that has been collected in database will automatically transfer to the new system. Also it is possible to purchase additional license for software product.

All versions include startup helper which will guide user how to work with software. Book guide included as well.

Free learning course is available for all registered clients.

Technical Support: is available for all registered clients.

Updates available every 3, 6, 12 months.

Systems requirements for 1C v 8:

Thin client: OS: Windows 7, Windows Server 2008 R2/2008/2003/2000, Windows Vista/XP/2000.

Web client OS: Windows 7, Windows Server 2008 R2/2008/2003/2000, Windows Vista/XP/2000, Linux, Mac OS x 10.5 and higher.

Web browser: Microsoft Internet Explorer 6.0, 7.0, 8.0 for Windows,

Mozilla Firefox 3.0-3.6. for Windows and Linux,

Google Chrome 4.0, 4.1 for Windows,

Safari 4.0.5 for Mac OS X.

It is support 5 DBMS : files, Oracle Database, Microsoft SQL server, Postgre SQL, IBM DB2.

Prices: “1C Accounting 8” Basic - 3300 rub. - (approx. 83 €)

“1C Accounting 8” Professional single license – 12600 rub. - (approx. 315 €)

“1C Accounting8” Professional. 5 licenses – 25500 rub. - (approx. 637 €)

“1C Accounting 8” Corporate – 28000 rub. - (approx. 700 €)

It is also possible to purchase “1C Accounting 8” in English language.

4.2. Overview of “Pohoda” software product

“Pohoda” – software product has been developed by software development company “Stormware”.

“Stormware” company: the Czech Software Development Company produces software products integrated with Microsoft Windows platforms. It operates in the Czech Republic and Slovakia. “Pohoda” represents applied solutions for small, medium and large sized business companies in area of enterprise resource planning, business and tax accounting.

Software products supported three languages: Czech and Slovak languages. Also interface has been translated in German and English languages.

“Pohoda” tracking documentation includes in two languages: Slovakian or Czech language.

System requirements:

OS: Windows 7, Vista SP1, Microsoft Windows XP SP3

Web client: Windows Server 2008, Windows Server 2003 R2.

CPU: Intel Core 2 Duo, 2 Ghz

RAM: 1X2048 mb, (For windows Vista 2X2049 mb.)

There are several configurations of “Pohoda” software which have different functionalities:

Table 4-2 Configurations of Software “Pohoda” (Source: Stormware Company)

Functions	Mini	Lite	Jazz			Standard			Profi		Premium			Complete		
	Pohoda	Pohoda	Pohoda	Pohoda SQL	Pohoda E1	Pohoda	Pohoda SQL	Pohoda E1	Pohoda	Pohoda SQL	Pohoda	Pohoda SQL	Pohoda E1	Pohoda	Pohoda SQL	Pohoda E1
Addressee	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Tax accountancy	*	*	–	–	–	*	*	*	–	–	–	–	–	*	*	*
Business accountancy	–	–	–	–	–	–	–	–	*	*	*	*	*	*	*	*
Financial Reports	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Home banking	–	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Taxes	*	*	–	–	–	*	*	*	*	*	*	*	*	*	*	*
Balance of payment	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Purchase orders	–	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

Accountability	–	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Foreign currency	–	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Estate property	–	*	–	–	–	*	*	*	*	*	*	*	*	*	*	*
Registry	–	*	–	–	–	*	*	*	*	*	*	*	*	*	*	*
Payroll	–	–	–	–	–	*	*	*	–	–	*	*	*	*	*	*
warehouse	–	–	*	*	*	*	*	*	–	–	*	*	*	*	*	*
Warehouses (additional functions)	–	–	–	–	*	–	–	*	–	–	–	–	*	–	–	*
Purchasing goods	–	–	*	*	*	*	*	*	–	–	*	*	*	*	*	*
Internet purchasing	–	–	–	–	*	–	–	*	–	–	–	–	*	–	–	*
Extension number of users	–	–	–	–	*	–	–	*	–	–	–	–	*	–	–	*
Managing access rights	–	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Access rights (additional functions)	–	–	–	–	*	–	–	*	–	–	–	–	*	–	–	*
Client-server technologies	–	–	–	*	*	–	*	*	–	*	–	*	*	–	*	*

Prices are:
1 license for 1 PC. Buyers can order additional license for additional money.

Pohoda 2011 Mini – 1980 CZK (81 €)

Pohoda Lite – 3890 CZK (159 €)

Pohoda Complete – 13980 CZK (570 €), SQL – 16780 CZK (685 €), E1 – 29980 CZK (1223 €)

“Stormware” Company offer also software product “Pohoda” for networks. Buyers can order version for 2-3 PC’s or 4-5 PC’s.

Updates are available 3 times per year for additional charge.

To get updates for the next year, users should buy new version of product.

From this table you may see that there is a wide range of configurations. The first configuration Pohoda 2011 “Mini” and “Jazz” can be used by individual business owners. Then when their business will growth they can upgrade to “Standard”, “Profi”, “Premium” or “Complete” version.

4.3. Characterization of potential user’s needs and requirements.

The first step for evaluation process is to specify user’s needs.
According to the report which has been done by Company Blytheco and Deloitte&Touch there are 10 top criteria for selecting accounting software. They ranked these criteria in order of importance. The results of survey have been split into two groups. First group was those who were in the process of searching accounting software for their business. The second group was those who were in the process of selecting their second system.

First time buyers:

Table 4-3 Rating of criteria for selecting accounting software according to the first time buyers opinion (Source: “Blytheco” Company)

Rank	Reason
1.	Price of software
2.	Ease of Implementation
3.	Ease of Use
4.	Software ability to fit the business
5.	Functionality of software
6.	Software works with existing hardware
7.	Growth potential of software
8.	Level of support provided by the local firm
9.	Quality of documentation
10.	Developers track record of performance

From the ranking table you can see that no one wants to overpay for software system and have problems with implementation or problems with using software product as well.
Here is a ranking table for the second time buyers:

Table 4-4 Rating of criteria for selecting accounting software according to the second time buyers opinion (Source: “Blytheco” Company)

Rank	Reason
1.	Level of support provided by the local firm
2.	Developers track record of performance
3.	Software ability to fit the business
4.	Growth potential of Software
5.	Price of Software
6.	Quality of documentation
7.	Functionality of software
8.	Ease of Use
9.	Ease of Implementation
10.	Software works with existing hardware

As you may see there are significant changes between these two tables.

Based on experience of second time buyers we may conclude that the most important role plays level of support provided by the local firm. The developer's track record from the last place moved to the second it means that it plays more important role that the first time buyers think. As we may see price of software product is not the most important but not the last one.

Obviously each type of business required different checklist for customer needs and requirements for accounting software systems according to business needs. For accountancy customers will need a software product which will allow them to prepare financial reports according to local regulations and for those who operates in different countries according to IFRS (International Financial Reporting Standards).

4.4. Comparisons of Characteristics and sub-characteristics.

Both of examined software products are specialized in the same area and have the same functionality.

Let's see what software product would have higher level of quality if we will compare prices for these products. Prices for basic versions almost the same but prices for corporative version of "1C Accounting 8" is equal to 700 € and for Complete E1 version of "Pohoda" is equal to 1223€. Conclusion according to the traditional way of comparisons software products would be following: "Pohoda" Complete E1 has higher level of quality than Corporate version of software "1C Accounting 8".

Evaluator should confirm or disprove this conclusion. For these purpose evaluator measures quality level for both products according to ISO 25010 methodologies.

Evaluation of quality characteristics and sub-characteristics has been done for Corporative version of "1C Accounting 8" and Complete version of "Pohoda 2011".

To measure characteristics of quality I used following structure [VANÍČEK, 2010]:

1) Name of measured attribute – a few words, which define attribute.
2) Name of measure.
3) Measure aim – question which is answered by the measure value.
4) Measure model – description of procedure, which define measure evaluation of attributes (measurement).
5) Data elements and formula for calculation of measure – description of data elements, which are input for the measure calculation.
6) Calculation values according to the formula.
7) Interpretation of value measure - definition of the range of possible measure value and our preference relation.
8) Measure scale type – defined used scale type.
9) Measure value type – by base value, for example number (breakdowns, mistakes, files, documents, functions) or time.
10) Data source for measure purpose – data source for measure (system specification, user's documentation, protocol about testing system, etc).
11) Lifecycle period, in which is able to measure – here is earliest lifecycle period, in which we have data needed for measure.
12) Professions using measure results – description of staffs, which will receive results of measurement.
13) Characteristics and sub-characteristics of quality substantially affected by this measure.
14) Characteristics and sub-characteristics of quality partially affected of measure.
15) Measure category – internal, external or for quality in use.

“1C accounting 8” VS “Pohoda” quality evaluation

Table 4-5 Evaluation of Reliability

№ of step	Information about sub-characteristic	
1	Reliability	
2	Failure density: shows how often failures occur per time unit. Main time between failures.	
3	Criteria: Is the software able to run for a long period of time without failures?	
4	For evaluation this criteria we should analyze data from distributor centers. Measure of number of failures has been taken from user’s report.	
5	Measurement function: $\frac{A}{t}$; A-number of failures, t-period of time (days) in which software program has been observed.	
6	1C Accounting 8 $\frac{16}{160} \approx 0.1$	Pohoda $\frac{7}{160} \approx 0.044$
7	Range of possible values: $0 \le x \le 1$. Less is better, 0 is best solution.	
8	Ratio scale type	
9	Number of breakdowns, mistakes, procedure failures occur per unit time	
10	Empirical way. Interview with representative from distribution centers of software products	
11	Lifecycle period: internal and external	
12	Developers, future users or acquirer	
13	Reliability: Fault tolerance, product maturity, Operability: user error protection	
14	Recoverability	
15	Internal attribute	

Table 4-6 Evaluation of Functional suitability

№ of step	Information about sub-characteristic	
1	Functional suitability	
2	Functional Implementation completeness	
3	Criteria: Can the software perform the tasks required?	
4	Total number of functions has been collected from documentation of both software products: 22	
5	Measurement function: $\frac{A}{B}$; A - number of functions realized in product. B - number of functions expected by end-users.	
6	"1C Accounting 8" $\frac{22}{22} = 1$	"Pohoda" $\frac{21}{22} = 0,955$
7	Range of possible values: $0 \le x \le 1$, greater is better, 1 is best solution	
8	Ratio scale type	
9	Number of functions	
10	Products documentation	
11	Lifecycle period: quality in use	
12	Future users	
13	Functional appropriateness,	
14	Operability: technical learnability, user error protection	
15	External attribute	

Table 4-7 Evaluation of Learnability

№ of step	Information about sub-characteristic	
1	Learnability	
2	Time length of specific courses to learn software product functionalities for effective using it.	
3	Criteria: Time needed for learning software functionalities	
4	Measurement based on period of time spending for learning functionalities of software product	
5	Measurement function: $\sum_{i=1}^{n} t_i$; t_i-number of hours required for learning specific course, i - specific course	
6	"1C Accounting 8" 64 academic hours	"Pohoda" 50 academic hours
7	Range of possible values: $0 \le x \le 120$; Less is better, 0 is best value.	
8	Absolute scale type	
9	Hours	
10	Training centers	
11	Lifecycle period: quality in use	
12	Future users, developers – to improve interactive design	
13	Usability: Ease of use, availability	
14	Operability: user interface aesthetics	
15	Quality in use	

Table 4-8 Evaluation of Interoperability

№ of step	Information about sub-characteristic	
1	Interoperability	
2	Interoperability on data level	
3	Criteria: Can the system interact with another system?	
4	Measurement based on number of data files, which have standard data format suitable for direct processing with other systems or number of functions, which can transfer data without problem	
5	Measurement function: $\frac{A}{B}$; A- number of successful cases/number of all cases, B-number of all cases.	
6	"1C Accounting 8" $\frac{267}{300} = 0.89$	"Pohoda" $\frac{14}{15} = 0.933$
7	Range of possible values: $0 \le x \le 1$; Greater is better, 1 is best solution	
8	Absolute scale	
9	Number of attempts to process data from experimental software product to another system	
10	Interview	
11	Lifecycle period: internal and external	
12	Future users, developers, auditors	
13	Compatibility: co-existence, Reliability: fault tolerance, recoverability	
14	Functional suitability: functional appropriateness	
15	External	

Table 4-9 Evaluation of Portability

№ of step	Information about sub-characteristic	
1	Adaptability	
2	Portability	
3	Criteria: Can the system interact with different OS platforms?	
4	Measurement based on number of OS. Into account have been taken most often used OS and ability to run application on PDA.	
5	Measurement function: $\frac{A}{B}$; A-number of platforms supported by experimental software program, B-total number of platforms most used in companies	
6	"1C Accounting 8" $\frac{11}{11}=1$	"Pohoda 2011" $\frac{5}{11}\approx 0,455$
7	Range of possible values: $0 \le x \le 1$, greater is better, 1 is best solution	
8	Ratio scale of type	
9	Number of supported OS platforms	
10	Product documentation	
11	Lifecycle period: internal and external	
12	Potential users	
13	Portability: Installability, Maintainability: reusability	
14	Maintainability: modularity	
15	External	

Table 4-10 Evaluation of Reliability

№ of step	Information about sub-characteristic	
1	System unavailability	
2	Reliability	
3	Criteria: How often software is not capable to run because of failures?	
4	Measurement based on observation the system for some period of time.	
5	Measurement Function: $\frac{A}{B}$; A- period of time, when system doesn't work because of failure/ period of time , B- period of time when running of system was required.	
6	"1C Accounting 8" $\frac{5}{160}=0,03125$	"Pohoda" $\frac{9}{160}=0,05625$
7	Range of possible values: $0 \le x \le 1$, less is better, 0 is best solution	
8	Ratio scale type	
9	Hours	
10	Interview	
11	Lifecycle period: Internal and external	
12	Potential users, developers, auditors	
13	Security: Confidentiality, Usability: user error protection, Reliability: fault tolerance, recoverability, Portability: replaceability.	
14	Usability: ease of use, learnability, user interface aesthetics, compatibility: interoperability	
15	External attribute	

Table 4-11 Evaluation of Performance efficiency

№ of step	Information about sub-characteristic	
1	Turnaround time	
2	Performance efficiency	
3	Criteria: How quickly does the system respond?	
4	Number of transaction per unit time	
5	Measurement Function: A; A - time for processing and saving document	
6	"1C Accounting 8" 1.75	"Pohoda" 2.08
7	Range of possible values: $0 \le x \le 60$; Less is better, 0 is best solution	
8	Ratio scale type	
9	Seconds	
10	Review from the internet test	
11	Lifecycle period: internal and external	
12	Potential users, auditors, developers	
13	Functional suitability: Functional appropriateness, Reliability: accessibility	
14	Maintainability: testability	
15	External	

Table 4-12 Evaluation of Analyzability

№ of step	Information about sub-characteristic	
1	Main failure analysis time	
2	Analyzability	
3	Criteria: Can faults be easily diagnosed?	
4	Average time spending for failure analysis	
5	Measurement Function: $\frac{A}{B}$; A-time for failure analysis per month, B-number of failures	
6	"1C Accounting 8" $\frac{2}{16} = 0.125$	"Pohoda" $\frac{3}{7} = 0.429$
7	Range of possible values: $0 \leq x \leq 1$, less is better, 0 is best solution.	
8	Ratio scale type	
9	Hours	
10	Interview	
11	Lifecycle period: external	
12	Potential users, developers	
13	Reliability: fault tolerance, Security: confidentiality, integrity, authenticity, non-repudiation, Usability: user error protection	
14	Usability: ease of use	
15	External	

Table 4-13 Evaluation of Security

№ of step	Information about sub-characteristic	
1	Control of access into system	
2	Security	
3	Criteria: Does the system controlled by administrator as required?	
4	A_1 - ability for managing access rights (0,25) A_2 - ability for delicate setting for each user (0,25) A_3 - password protection (0,25) A_4 -reports about last changing in the system (0,25)	
5	Measurement Function: $\sum_{i=1}^{4} A_i$; A_i - method of security	
6	"1C Accounting 8" $A_1 + A_2 + A_3 + A_4 = 1$	"Pohoda" $A_1 + A_2 + A_3 = 0,75$
7	Range of possible values: $0 \le x \le 1$, greater is better, 1 is best solution.	
8	Ordinal scale type	
9	Methods of security	
10	Documentation	
11	Lifecycle period: internal and external	
12	Potential users	
13	Reliability: fault tolerance, Security: confidentiality, integrity, authenticity, non-repudiation, Usability: user error protection	
14	Portability: installability, reliability: availability	
15	External	

4.5. *Discussion*

Based on evaluation of quality characteristics and sub-characteristics we may see that some of sub-characteristics have better values for "1C Accounting 8" software, but another for "Pohoda".

We may see all results in the following table:

Table 4-14 Quantitative values of software quality characteristics (Author's calculation)

Value of quality characteristic, sub-characteristic	"1C Accounting 8"	"Pohoda"
1) Functional suitability	Max	Max
	1	0,955
2) Reliability (Failure Density)	Min	Min
	0,1	0,044
3) Reliability (System unavailability)	Min	Min
	0,03125	0,05625
4) Performance efficiency	1,75	2,08
5) Usability (Learnability)	Min	Min
	64	50
6) Security	Max	Max
	1	0,75
7) Compatibility (Interoperability)	Max	Max
	0,89	0,933
8) Maintainability (Analyzability)	Min	Min
	0,125	0,429
9) Portability (Adaptability)	Max	Max
	1	0,455

Evaluator can easily compare values for each characteristic pairwise. Evaluation process would be finished when expert will rate each of software products.

A fuzzy simple additive weighting method (SAW) has been used in order to obtain the final evaluation grade for each software product.

First step is summarizing of all evaluation criteria values for different attributes belongs to the same characteristic.

In this paper two attributes for ’Reliability’ characteristic had been measured. According to the standard model of Quality measures ‘Reliability’ has the weight 0,3. In our case each attribute obtains the weight 0,15.

2 step. Input all data into Table Decision Matrix:

Table 4-15 Decision matrix (Author’s calculation)

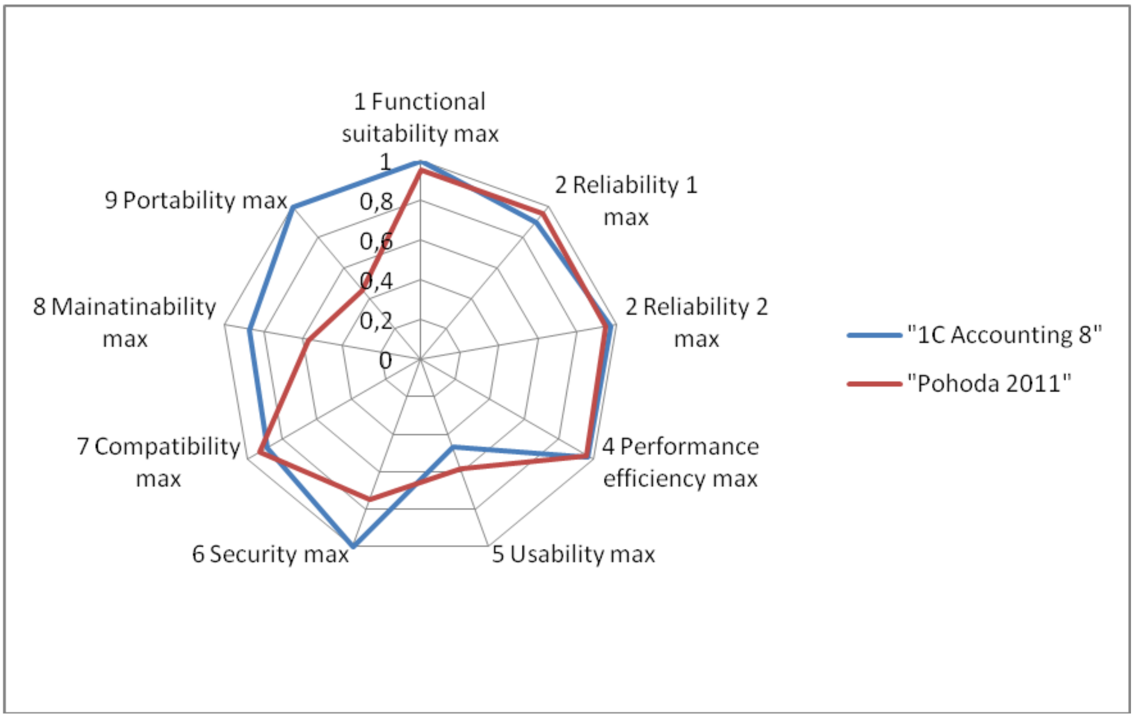
	Quality Characteristics								
	1	2	3	4	5	6	7	8	9
	Functional Suitability	Reliability (FD) 1	Reliability (SU) 2	Performance Efficiency	Usability	Security	Compatibility	Maintainability	Portability
Criteria weight									
Criteria characteristic	Max	Min	Min	Min	Min	Max	Max	Min	Max
“1C Accounting 8”	1	0,1	0,03125	1,75	64	1	0,89	0,125	1
“Pohoda”	0,955	0,044	0,05625	2,08	50	0,75	0,933	0,429	0,455

Some of criteria characteristic are Max, but other criteria characteristic are Min. Moreover while evaluation process for quality characteristics had been used different scales. To continue calculation all values should be transformed to one scale and normalized.

After normalization process and transformation all criteria to Max Decision Matrix will obtain following values:

Table 4-16 Normalized and Transformed Decision Matrix (Author’s calculations)

	Quality Characteristics								
	1	2	3	4	5	6	7	8	9
	Functional Suitability	Reliability (FD) 1	Reliability (SU) 2	Performance Efficiency	Usability	Security	Compatibility	Maintainability	Portability
Criteria weight									
Criteria characteristic	Max	Max	Max	Max	Max	Max	Max	Max	Max
“1C Accounting 8”	1	0,9	0,969	0,971	0,467	1	0,89	0,875	1
“Pohoda”	0,955	0,965	0,944	0,965	0,583	0,75	0,933	0,571	0,455

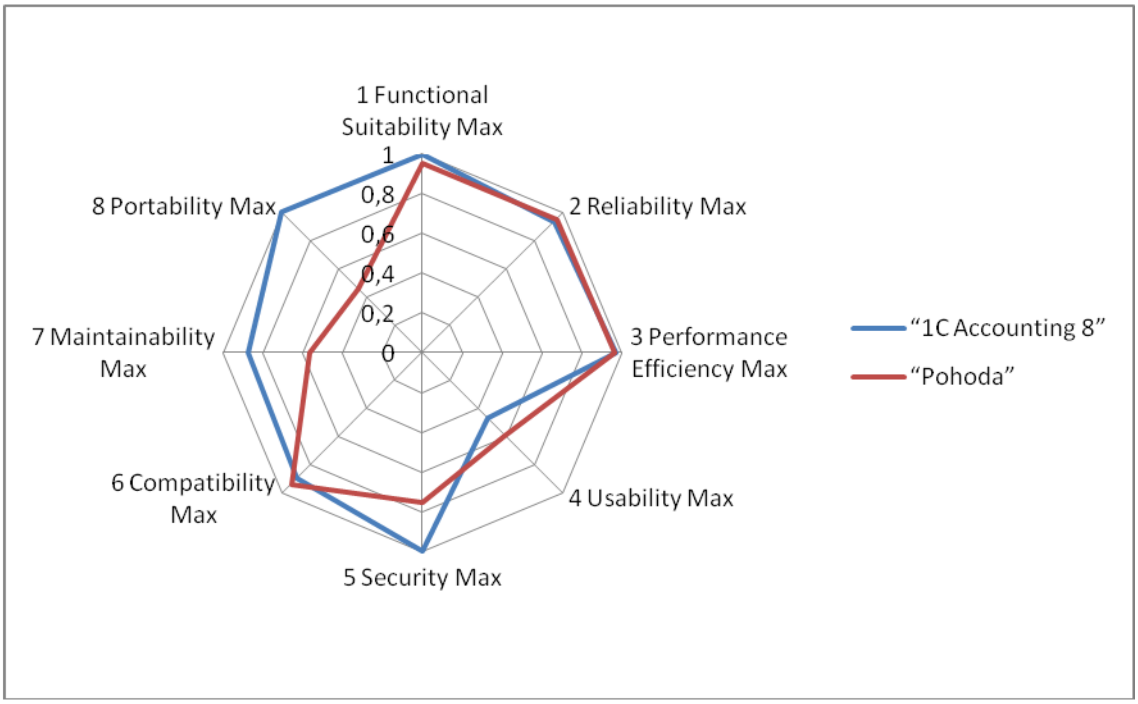


Figures 4-1 Radar diagram. Graphical representation of normalized and transformed matrix

After normalization and transformation process total rating for reliability calculates:

Table 4-17 Decision matrix for quality characteristics

	Quality Characteristics							
	1	2	3	4	5	6	7	8
	Functional Suitability	Reliability	Performance Efficiency	Usability	Security	Compatibility	Maintainability	Portability
Criteria weight								
Criteria characteristic	Max	Max	Max	Max	Max	Max	Max	Max
"1C Accounting 8"	1	0,934	0,971	0,467	1	0,89	0,875	1
"Pohoda"	0,955	0,950	0,965	0,583	0,75	0,933	0,571	0,455



Figures 4-2 Radar diagram. Graphical representation of decision matrix

In this graph all vertixes are represent quality characteristics. Evaluator could see that "1C Accounting 8" has higher level of quality than "Pohoda". To make sure evaluator continues evaluation procedure in order to obtain numerical value for quality software products.

For the next step has been calculated weight (degree of importance) for each criterion and score method has been used for criteria assessment. In this document values for weight criteria had been taken from standard model of Quality measures. All measures have been transformed to the new scale for convenience:

Values from standard model of Quality measures: Functional suitability – 0,3; Reliability – 0,3; Performance efficiency – 0,1; Usability – 0,1; Security – 0,2; Compatibility – 0,15; Maintainability – 0,1; Portability – 0,1. Range of possible values [0;1].

After transformation has been obtained following values: Functional suitability – 0,22; Reliability – 0,22; Performance efficiency – 0,07; Usability – 0,07; Security – 0,15; Compatibility – 0,11; Maintainability – 0,07; Portability – 0,07. Range of possible values [0;1].

After evaluation of all characteristics using fuzzy simple additive weighting method have been calculated final grades for both software products:

Quality rating for Corporate version of "1C Accounting 8" is equal to 0,593.

Quality rating for Complete E1 version of "Pohoda" is equal to 0,407.

5. Conclusion

The data used for evaluation quality characteristics of software products for accounting are strongly influenced by subjective reasons: conditions for measurements were different; some of data have been obtained based on experience of different users. Evaluator established criteria for evaluation by himself. Evaluator decides which criteria would be more effective to obtain quality rating with high level of accuracy. There were a lot of limitations during evaluation process; it was not possible to evaluate some of sub-characteristics because of access limitation to the information, access limitation to the software products. Evaluator should establish more evaluation criteria for each characteristic and sub-characteristic to get more accuracy quality rating software products.

To minimize subjective influences and limitations evaluation process has been done according to the recommendations of standard ISO 25010 [ISO, 2010]. Corporate version of "1C Accounting 8" and Complete E1 version of "Pohoda" software had been evaluated. From the decision matrix the results are following: "Pohoda" Complete E1 has three out of eight quality characteristics more suitable than Corporate version of "1C Accounting 8" has: 'Usability', 'Reliability' and 'Compatibility'. Another five quality characteristics: 'Functional suitability', 'Performance efficiency', 'Security', 'Maintainability', 'Portability'; are greater are more suitable for "1C Accounting 8".

The overall quality level has been measured by fuzzy simple additive weighting method. Rating scale from zero to one has been used for evaluation, where 0 stands for the worst and 1 stands for the best results. According to this method the quality rating for Corporate version of "1C Accounting 8" is 0,593 and the quality rating for Complete E1 version of "Pohoda" software is 0,407. It indicates that software "1C Accounting 8", Corporate version has higher level of quality than software "Pohoda", Complete E1 version has.

In addition, I conclude that traditional way of comparing software products quality according to the level of products prices is not working in reality.

6. References

1. BOEHM B. W., BROWN J.R., KASPAR H., LIPOW M., McLEOD G., MERRIT M. [1978]. Characteristics of Software Quality. North Holland Publishing, Amsterdam, The Netherlands.
2. BOEHM B.W., BROWN J.R., LIPOW M. [1976]. Quantitative evaluation of software quality. In Proceedings of the 2nd international conference on Software Engineering. IEEE Computer Society, Los Alamitos (CA), USA; 592 – 605.
3. DIRLEWANGER Werner [2006]. Measurement and Rating of Computer Systems Performance and of Software Efficiency. An introduction to the ISO/IEC 14756 Methods and a Guide to its Application, Kassel University press GmbH, ISBN-10: 3-89958-233-0, ISBN-13: 978-3-89958-233-0.
4. DOMEOVÁ L. [2010]. Decision Support Systems, lectures. CZU.
5. DROMEY R.G. [1995]. A model for software product quality. IEEE Transactions on software Engineering; 21: 146-162.
6. DROMEY R.G. [1996]. Concerning in Chimera [software quality]. IEEE Software; 13:33 – 43.
7. GRADY R.B. [1992]. Practical Software Metrics for Project management and Process Improvement. Prentice Hall, Engelwood Cliffs, NJ, USA.
8. ISO 9000 [2005]. Quality management systems – Fundamentals and vocabulary. International Organization for Standardization, Geneva, Switzerland.
9. ISO/IEC IS 9126 [1991]. Software Product Evaluation – Quality Characteristics and Guidelines for their Use. International Organization for Standardization, Geneva, Switzerland.
10. ISO/IEC 9126 – 1 [2001]. Software Engineering – Product quality – Part 1: Quality model. International Organization for Standardization, Geneva, Switzerland.
11. ISO/IEC 9126 – 2 [2003]. Software Engineering – Product Quality – Part 2: External metrics. International Organization for Standardization, Geneva, Switzerland.
12. ISO/IEC 9126 – 3 [2003]. Software Engineering – Product Quality – Part 3: Internal metrics. International Organization for Standardization, Geneva, Switzerland.

13. ISO/IEC 9126 – 4 [2004]. Software Engineering – Product Quality – Part 4: Quality in use metrics. International Organization for Standardization, Geneva, Switzerland.
14. ISO/IEC 12207 [2008]. System and software engineering – software life cycle processes.
15. ISO/IEC 14598 [1999]. Software engineering – Software product evaluation.
16. ISO/IEC 25010 [2010]. Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – System and software quality models.
17. ISO/IEC 25020 [2010]. Software engineering - Software product quality requirements and evaluation (SQuaRE) – Measurement reference model and guide.
18. JACOBSON I. BOOCH G, RUMBAUGH J. [1999]. The Unified Software Development Process. Adisson Wesley.
19. KITCHENHAM B., PFLEEGER S.L., [1996]. Software Quality: the Exclusive Target. IEEE Software; 13: 12-21.
20. KRUCHTEN P. [2000] The Rational Unified Process: An introduction. Addison Wesley.
21. McCALL J.A., RICHARDS P.K., WALTERS G. F. [1977] Factors in software quality, Volumes 1, 2 and 3 US Rome Air Development Center Reports, US department of Commerce, USA.
22. MOSER Simon [1996] Measurement and Estimation of Software and Software processes, Bern.
23. RAFA E.Al- QUITASH [2010] Quality models in Software Engineering Literature: An analytical and comparative study. Journal of American Science; 6(3), 166-175
24. STRUSKA Zdeněk, VANÍČEK Jiří, [2010]. Measurement and rating of information systems quality. Part 3: Design complexity and software engineering consequences, CZU.
25. VANÍČEK Jiří, [2010]. Measurement and rating of information systems quality. Part 1: Concepts, terminology and theoretical back ground, CZU.
26. VANÍČEK Jiří, [2010] Information systems quality rating, CZU, ISBN 978-80-213-2062-8

27. VANÍČEK Jiří, VRANA Ivan, STRUSKA Zdenek, [2010]. IT product quality, impact on e-governance, measurement and evaluation.

Web links:

28. “1C” Company, Development software Company in Russia, Information about software product “1C Accounting 8”. Available from WWW:
< http://v8.1c.ru/buhv8/ >
29. Blytheco Company, Top ten criteria for selecting accounting software. Available from WWW:
<http://www.blytheco.com/pdf/misc/top_ten_criteria_for_selecting_software.pdf >
30. LOZININ A.I. SUBINSKY I.B., Quality characteristics of software and measurement methods. Available from WWW:
< http://www.ibtrans.ru/Estimating%20methods.pdf >
31. “Stormware”. Development software Company in the Czech Republic, Information about accounting software “Pohoda”. Available from WWW:
< http://www.stormware.cz/ >
32. Wikipedia: definition of Accountancy, available from WWW:
< http://en.wikipedia.org/wiki/Accountancy >
33. Wikipedia: definition of accounting software. Available from WWW:
< http://en.wikipedia.org/wiki/Accounting_software >