MASTER'S THESIS

EXPERIMENTING EWAM METHOD TO EVALUATE THE E-COMMERCE WEB SITE QUALITY

Studiengang

Informationstechnologien und Wirtschaftsinformatik

By Maria-Aurora Cuzma

Registration number: 1510320005

Graz, July 2018

STATUTORY DECLARATION

I declare that I have developed and written the enclosed Master Thesis completely by myself, and have not used sources or means without declaration in the text. Any thoughts from others or literal quotations are clearly marked. The Master Thesis was not used in the same or in a similar version to achieve an academic grading or is being published elsewhere.

Graz, July 2018

Maria-Aurora Cuzma

ACKNOWLEDGEMENT

I would first like to thank my thesis advisor DI Michael Neuwersch of the Campus02 Graz. He consistently supported me to finish this paper and he steered me in the right direction whenever he thought I needed it.

I would also like to acknowledge all of my Teachers of the Campus02. I am gratefully for their very valuable knowledge and for their experiences they shared with me on the study time.

And finally, I would like to thank my family, I couldn't finish this study without their support.

Abstract

The development of the information society, the increase in the number of users make the transmission of information more diverse but also attractive as possible and easier to use. Under these circumstances, public or private institutions choose to synthesize and share information through the web sites. More, online shops are nowadays very popular and used because people want to save time, money and flexibility of payment methods.

Recognizing the importance of software products quality has led to key changes in the behavior of software producers by their focus on delivering solutions to ensure client satisfaction. In order to develop software products with a high level of quality and client satisfaction, software developers have adopted best practices and standards for quality management in the software product life cycle. The results of numerous European projects as well as international quality standards have been and are still being adopted and capitalized through quality improvement programs.

To analyze and compare the quality of web sites to see if they meet the users' requirements are proposing to develop a methodology based on the procedures and algorithms used in the method EWAM - The Extended Web Assessment Method, a methodology that will end with a series of recommendations on designing or implementing web sites. The method responds to a major problem related to the evaluation of e-commerce applications: which are the most relevant criteria to ensure the success of an e-commerce system.

EWAM is a complex method that was created for a detailed analysis of a sample of applications and websites in a particular sector. In this respect, the method is not intended for the mass evaluation of websites.

The EWAM method uses the specific "profile" concept. This is a reference against which the results from evaluating an e-commerce application are compared. One of such profiles have been defined *"Best practice" profile* – which means a set of methods, techniques, procedures, etc. demonstrated and proven to be the best in current practice and used in the provision of e-commerce services.

The evaluation criteria are formulated in general terms and are valid for any sector, but they are differentiated by the sector-specific importance. In this way, the conditions for identifying the reference sector and benchmarking of different e-commerce applications are created both within a sector and between different sectors.

The establishment of (referential) “profiles” allows the comparison of the quality of an e-commerce application with the average quality of a given sector, and, at the same time, the comparison with the best practices used in the sector.

The focus lies on consumer perspectives and the specific features of the Internet as a medium. Using the EWAM tool, an analysis of four commercial Web sites in the same business sectors – fashion but in two different countries, will be performed: the websites as Orsay (www.orsay.com), Zara (www.zara.com), H&M (www.hm.com) and Mango (www.mango.com). The results will be per country compared. Furthermore, the work analyses the conceptual elements such as quality models, software product quality in general and the methodology to prove the web sites quality as well - The Extended Web Assessment Method.

The empirical section based on The Extended Web Assessment Method examines and evaluate the online shops of Orsay, Zara, H&M and Mango websites based on interviewing people from Austria and Romania. Web assessment is a very ambitious and labor-intense work. The assessors should meet certain criteria:

- They need to understand the criteria of the Web assessment form very well, hence they must undergo a thorough instruction
- They must be experienced Web users
- They must take the time to go through all four transaction phases for each Web sites assessed (including delivery and payment!)

The objectives of experimenting with the EWAM method are:

- verifying and validating website evaluation criteria specified in the EWAM method;
- verifying the applicability of the method for measuring and evaluating websites in the field of fashion e-commerce in Romania and Austria - its comparing;
- formulating new criteria on the evaluation of websites, in general, and of fashion e-commerce websites, using results comparison from two European countries – Austria (central Europe) and Romania (eastern Europe)

Additionally, a set of questions will be performed to find if e-commerce systems, especially B2C systems in fashion sector, operate in selected target countries as well as it does in its home country, across geographic boundaries. Does the site respect cultural differences between the home and foreign country? Which are the particularly needs to for fashion e-commerce websites? Are the user needs changed due modern technology – use they nowadays more web or apps? The answers will conclude if the ‘best practices’ profiles can be extended with new criteria on evaluation of fashion e-commerce websites due modern technology and internationalization/globalization.

Contents

1. INTRODUCTION

The development of the information society, the increase in the number of users make the transmission of information more diverse but also attractive as possible and easier to use. Under these circumstances, public or private institutions choose to synthesize and share information through the web sites. More, online shops are nowadays very popular and used because people want to save time, money and flexibility of payment methods.

Recognizing the importance of software products quality has led to key changes in the behavior of software producers by their focus on delivering solutions to ensure client satisfaction. To develop software products with a high level of quality and client satisfaction, software developers have adopted best practices and standards for quality management in the software product life cycle. The results of numerous European projects as well as international quality standards have been and are still being adopted and capitalized through quality improvement programs.

To analyze and compare the quality of web sites to see if they meet the users' requirements are proposing to develop a methodology based on the procedures and algorithms used in the method EWAM - The Extended Web Assessment Method, a methodology that will end with a series of recommendations on designing or implementing web sites. The method responds to a major problem related to the evaluation of e-commerce applications: which are the most relevant criteria to ensure the success of an e-commerce system.

EWAM is a complex method that was created for a detailed analysis of a sample of applications and websites in a particular sector. In this respect, the method is not intended for the mass evaluation of websites.

The EWAM method uses the specific "profile" concept. This is a reference against which the results from evaluating an e-commerce application are compared. One of such profiles have been defined *"Best practice" profile* – which means a set of methods, techniques, procedures, etc. demonstrated and proven to be the best in current practice and used in the provision of e-commerce services.

The evaluation criteria are formulated in general terms and are valid for any sector, but they are differentiated by the sector-specific importance. In this way, the conditions for identifying the reference sector and benchmarking of different e-commerce applications are created both within a sector and between different sectors.

The establishment of (referential) "profiles" allows the comparison of the quality of an e-commerce application with the average quality of a given sector, and, at the same time, the comparison with the best practices used in the sector.

The focus lies on consumer perspectives and the specific features of the Internet as a medium. Using the EWAM tool, an analysis of four commercial Web sites in the same business sectors – fashion but in two different countries, will be performed: the websites as Orsay (www.orsay.com), Zara (www.zara.com), H&M (www.hm.com) and Mango (www.mango.com). The results will be per country compared. Furthermore, the work analyses the conceptual elements such as quality models, software product quality in general and the methodology to prove the web sites quality as well - The Extended Web Assessment Method.

The empirical section based on The Extended Web Assessment Method examines and evaluate the online shops of Orsay, Zara, H&M and Mango websites based on interviewing people from Austria and Romania. Web assessment is a very ambitious and labor-intense work. The assessors should meet certain criteria:

- They need to understand the criteria of the Web assessment form very well, hence they must undergo a thorough instruction
- They must be experienced Web users
- They must take the time to go through all four transaction phases for each Web sites assessed (including delivery and payment!)

The objectives of experimenting with the EWAM method are:

- verifying and validating website evaluation criteria specified in the EWAM method;
- verifying the applicability of the method for measuring and evaluating websites in the field of fashion e-commerce in Romania and Austria - its comparing;
- formulating new criteria on the evaluation of websites, in general, and of fashion e-commerce websites, using results comparison from two European countries – Austria (central Europe) and Romania (eastern Europe)

Additionally, a set of questions will be performed to find if e-commerce systems, especially B2C systems in fashion sector, operate in selected target countries as well as it does in its home country, across geographic boundaries. Does the site offer the option to view non-English pages? Does the site respect cultural differences between the home and foreign country? Which are the needs to for fashion e-commerce websites? Are the user needs changed due modern technology – use they nowadays more web or apps? The answers will conclude if the ‘Best practices’ profile can be extended with new criteria on evaluation of fashion e-commerce websites due modern technology and internationalization/globalization.

Hypotheses and research question

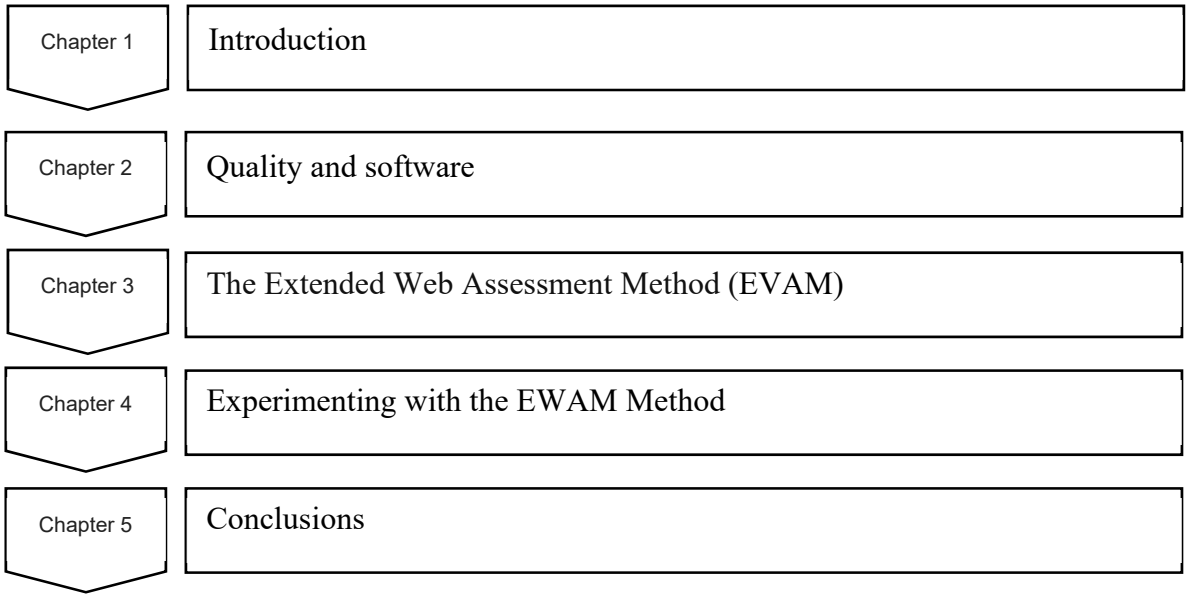
Which new criteria for the “Information” phase on the evaluation of fashion websites extends ‘Best practices’ profile to improve the quality of e-commerce web site?

Research hypothesis:

H1: The new criteria for the “Information” phase on the evaluation of fashion websites extends ‘Best practices’ profile to improve the quality of e-commerce web site

H0: The new criteria for the “Information” phase on the evaluation of fashion websites doesn’t extend ‘Best practices’ profile to improve the quality of e-commerce web site

Works construction



To answer the question of how software quality can be guaranteed, we must first define the term of quality and, especially, software quality. To do this, in Chapter 2 we analyzed quality concepts and philosophy, quality models and the conceptual-methodological framework of the quality of software products.

EWAM (**E**xtended **W**eb **A**ssessment **M**ethod) will be presented in chapter 3 and the quantitative study, the explanation, results and answers interpretation will be presents in Chapter 4.

Chapter 5 presents the final conclusions.

2. QUALITY AND SOFTWARE

Modern society uses the discoveries of the computer science in almost all daily situations. This omnipresence of computers in all of their versions leads to electronic systems connections and thus to an increase in total complexity. Since many years the percentage of software components has already been exceeding the percentage of hardware components and most of the systems functions cannot be achieved without software and software has become the number one source of errors (Hoffmann, 2008, p. 6). In addition to the complexity requirement, this is an essential element of why we must strive for high quality, this also accounts for the potential economic advantage to be achieved by the competition.

2.1. Quality concepts and philosophies

Philip B. Crosby thoroughly analyzes this topic in his book “Quality is free”. With its end “Quality is ballet, not hockey.” (Crosby, 1979, p.15), he relativizes the impetuous hopes of a direct and naive solution to the problem. He covers four false hypotheses to explain to the reader his point of view on quality. Crosby sees as a first mistake the hypothesis that quality is absolute. He refers to the fact that each individual speaking about quality uses an own subjective standard and relative-relation system. This prompted Crosby to resolve another hypothesis, namely that quality is not tangible and cannot be measured. He also abolishes the hypothesis that economic framework conditions, such as quality costs, would be relevant and a higher quality from a product / service could be achieved. Resolving the last bias on his view on quality – namely that quality problems are found primarily with employees and in the manufacturing sector – he finally arrives at the statement that “Quality is meeting requirements” (Crosby 1979, p. 6).

Joseph M. Juran developed in his book “Quality by Design” a philosophy of companies stating that quality is manifested in three steps that are repeated within a company. These are quality planning, quality regulation and, ultimately, quality improvement, the so-called Juran trilogy (Juran, 1992, p.14-21). Juran also coined the expression “vital few, useful many”, renders an application of the Pareto principle (the 80-20 rule) on quality assurance, which mainly supports the fact that most relevant effects are based on a relatively small number of causes. Another known mentor in quality sciences, Kaoru Ishikawa, summarizes his knowledge on quality by the slogan “Quality first” and implements quality and quality management firstly in management. For Ishikawa also, consumers define quality and companies must strive to meet these requirements by involving all departments within the company.

Garvin classifies in *Managing quality* the term quality after evaluating an empirical study in five hypotheses he explains in his work as follows:

The "transcendent" hypothesis

According to this way of thinking, quality is inherent to a product or service and is a high and comprehensive requirement for the functionality of a product (Garvin, 1988, p.40-42). Representatives of this hypothesis are of the opinion that quality cannot be measured or defined precisely in this purity.

The hypothesis that relates to the product

According to Garvin (1988, p.42), differences in quality are based on the different characteristics of the criteria measured, desired or expected from the product analyzed.

The hypothesis related to the user

Garvin (1988, p.43) argues that quality is less established by the actual product than by the user of the product. Garvin also states that different users have different needs and desires related to the product performance factors. The product that meets these different needs will have a good quality characteristic.

The hypothesis related to the process and the manufacturer

According to Garvin (1988, p. 44) the competitive advantage of cost reduction in this hypothesis is important and refers to (Crosby, 1979, p.18), defining quality in this hypothesis as "conformance to requirements".

The hypothesis reported to price and utility

This hypothesis goes further by analyzing quality compared to the hypothesis related to process and manufacturer. According to Garvin (1988, p.45), a quality product is one that meets the desired utility and simultaneously complies with the requirements defined, but at acceptable costs for potential users.

2.2. Quality models

A quality model attempts to classify and specify the quality term in all its individual aspects. This usually happens by dividing logically the characteristics of quality and the partial characteristics which are divided into quantifiable indicators. While an atomic indicator can be measured and determined, the same rule does not apply to quality characteristics.

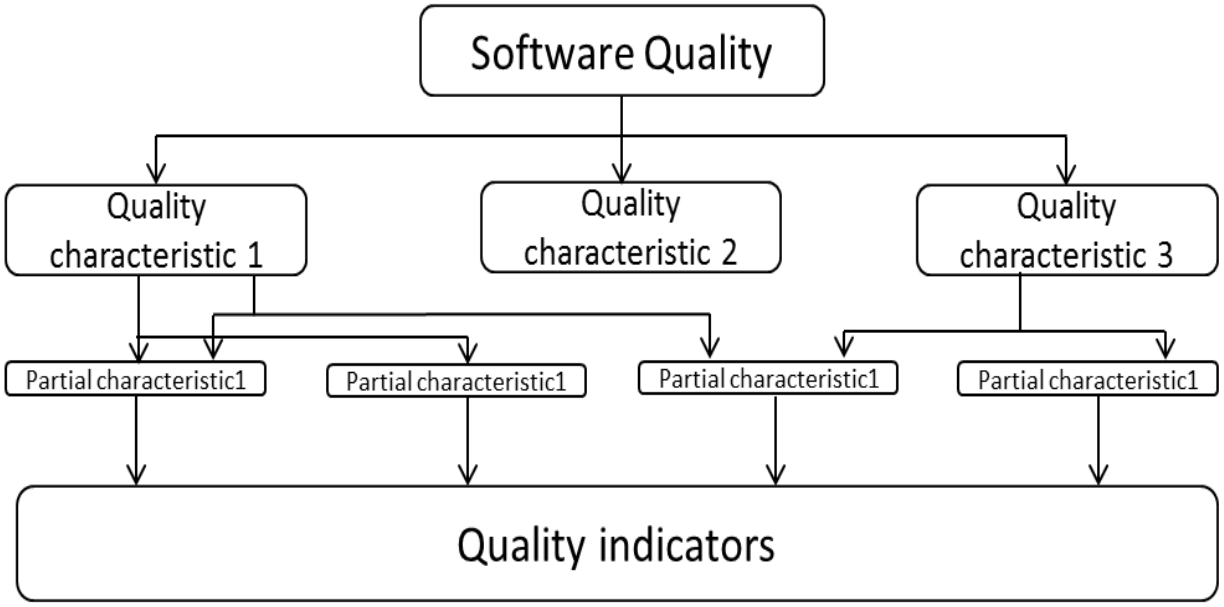


Figure 1 The structure of quality models (Balzert, 1998, p. 257)

The classification of general quality characteristics of products, processes and services, and the analysis of the special features of software represent the basis for assessing the quality of software in particular.

It is not only the correct requirements that have to be achieved, because quality mixes relative and subjective aspects. The quality concepts and philosophies discussed above are more quality oriented while the quality models presented below also state quantitative aspects.

2.2.1. The McCall quality model

One of the first models to measure software quality is the McCall, Richard & Walters model, in which the authors identify 55 candidates for quality factors, reducing and classifying these in 11 quality factors through a process of evaluation. Initially, the quality model was developed for the US Department of Defense and aimed to improve understanding between users and developers.

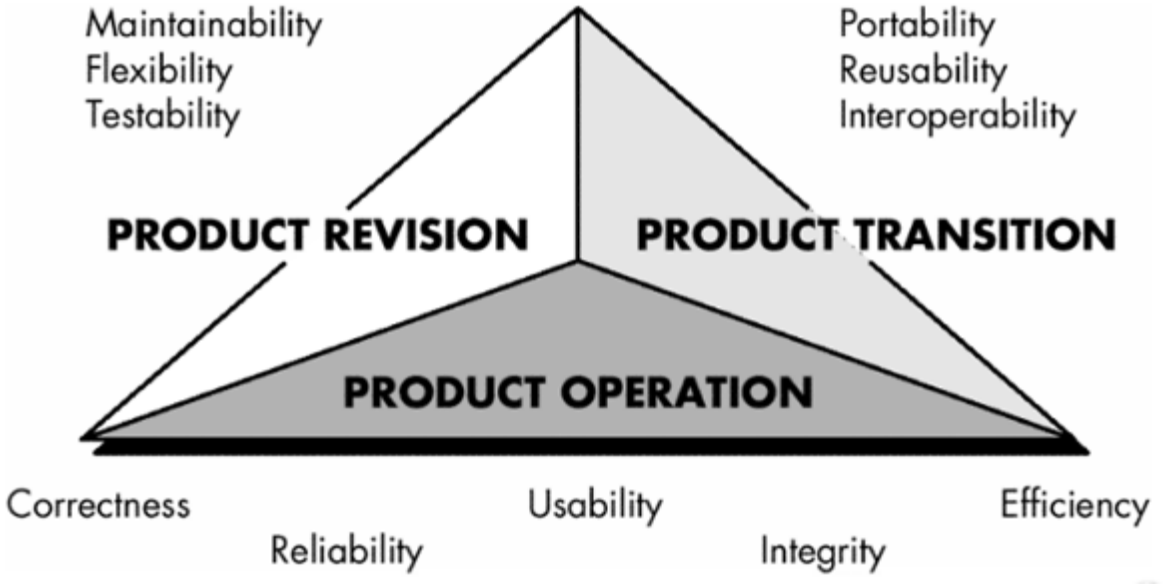


Figure 2 McCall's Triangle of Quality (McCall. Richards & Walters, 1978)

McCall, Richards & Walters allocate the quality factors analyzed in three areas where they play a special role. As shown earlier, the class "review" is characterized by the following questions: "Can the problem be solved?", "Can changes be made easy?", and "Can the code be tested?". Typical issues in the "Product transition" class are: "Is the code operational in a different running environment?", "Are parts of the code reusable?", and "Is the system able to communicate with other systems?". Questions related to operation, reliability and ease of use are found in the "Product operations" class. Thus this model takes into account both the priorities of users and of developers. The quality characteristics are listed in Appendix 2.2.

2.2.2. Boehm's quality model

Boehm (1978) improved the McCall model and has built it hierarchically introducing three distinct levels of abstraction.

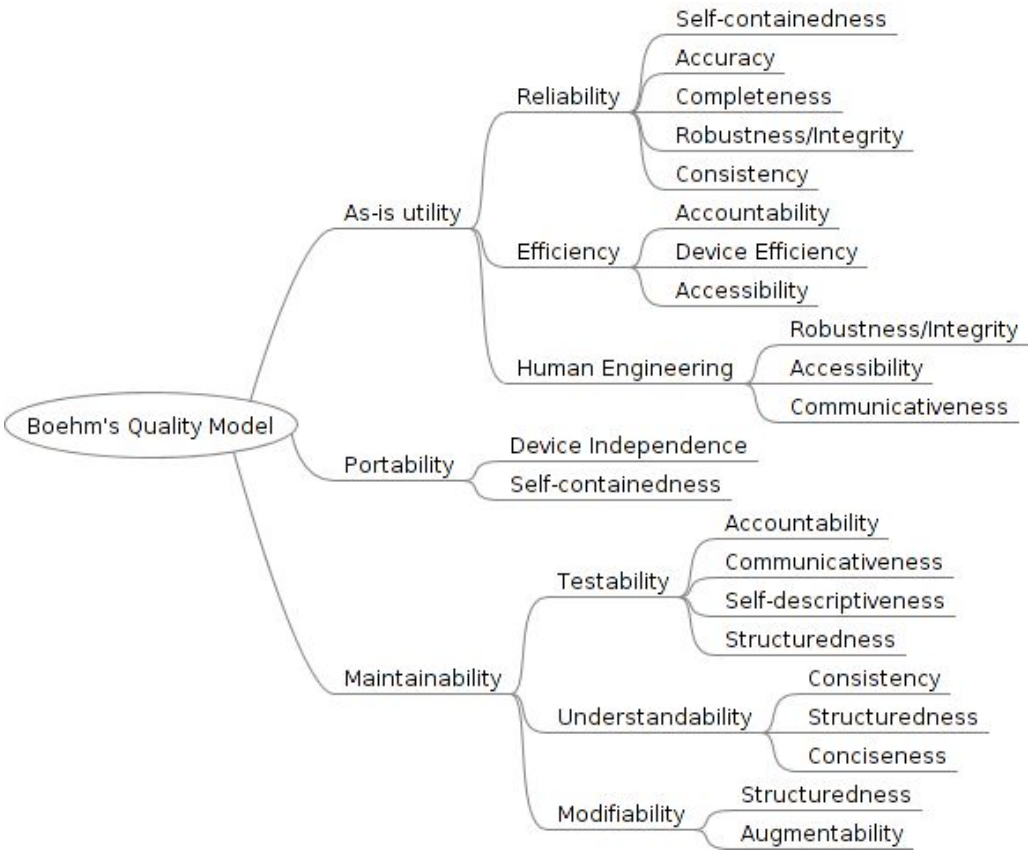


Figure 3 Boehm's quality model (Boehm, 1978)

The highest-level combines characteristics such as efficiency, ease of use, maintenance and portability, placing the user in focus.

As-is Utility

The overall usefulness of the product in terms of simplicity, efficiency and certainty. Boehm's definitions in this criteria category are basically the same as those identified by McCall and we will not detail them more at this point. The classification criteria "Human engineering" can be best compared with McCall's Usability criteria category, although they are not identical.

Maintainability

This criteria category includes the ability to test, develop the product, and simplify use.

Portability (General Utility)

This criteria category defines the effects on the product when changing the work environment.

A quality model is a structured collection of characteristics of a software system to systematically establish the software's quality. It can be recognized that the three quality models analyzed present a complete overlapping in the case of five characteristics, and a partial overlapping in the case of three characteristics. Four characteristics are analyzed only in one model.

2.3. The Conceptual-Methodological Framework of the Quality of Software Products

Software products have become essential components of many systems and products, as well as a new major factor in the trade of products and services. Furthermore, with the new global quality requirements, international agreements on the procedures for the evaluation of the quality of software products are becoming more and more important.

The complexity and multitude of factors that influence the quality of software products require multidisciplinary research, and the application of modern research methods that have demonstrated their usefulness and effectiveness in practice. Improving the quality of software products is one of the main forms in which creativity is materialized.

The economic value of a software product results from how its quality is perceived by clients or end users. Quality is more and more perceived as a critical attribute of the software product because the lack of quality leads to user dissatisfaction, financial loss, and even loss of life.

Worldwide, research and achievements in the quality of software products are at an advanced stage. They have been accelerated by key factors such as increasing international competition, high rates of change in software and hardware technologies, new client requirements, requirements related to standards compliance, and so on.

Recognizing the importance of software products quality has led to key changes in the behavior of software producers by their focus on delivering solutions to ensure client satisfaction.

In order to develop software products with a high level of quality and client satisfaction, software developers have adopted best practices and standards for quality management in the software product life cycle.

Information technology companies, irrespective of their size, have developed long-term strategies and programs in which ***product*** quality management is integrated with the quality management of design and development ***processes*** (see, for example, Côté et al., 2004; Jalote, 2002; Jones, 2000).

The results of numerous European projects as well as international quality standards have been and are still being adopted and capitalized through quality improvement programs. Software product quality improvement programs have been implemented in companies that have significant financial resources and are designed to maintain competitiveness and / or provide a competitive edge on the product and service market.

The software product quality field has been approached by many researchers, academics, and organizations in the software industry. Research has mainly addressed methodological and experiential aspects by taking over and adapting results from internationally prestigious firms.

The main objective of this chapter is to present to the readers interested in software product quality a conceptual and methodological framework based on international standards. The conceptual framework helps organizations and staff in these organizations with responsibilities in specifying quality requirements, measuring and evaluating information technology products and systems.

2.3.1. Standards on the quality of software

2.3.1.1 ISO Standards Series 9126 and Series 14598

In 1985, ISO / IEC JTC1 SC7 / WG61[1] began the development of the international standard ISO 9126, based on the results of the research obtained during 1977-1980 by a number of specialists in the field (McCall, Boehm, etc.). The first step of the ISO Technical Committee to systematically organize the properties of software products has failed due to different definitions and interpretations.

SC7 / WG6 experts have decided that the best solution for setting an international standard for software product quality is to recommend a set of features based on a definition of quality that was later used in ISO 8402:1994 Quality Management Systems – Vocabulary. The definition has been accepted for all types of products and services, starting from the user's needs.

The main result of the SC7 / WG6 activity was a set of six software product quality features. However, a terminology standard containing definitions of features would not provide sufficient support for users to evaluate the quality of software products. That is why a description of the quality evaluation process has been included.

In practice, product quality evaluation also requires features other than those in the set considered, as well as metrics for each of the features. The state of development in

[1] The International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC) Joint Technical Committee 1 (JTC1) Subcommittee 7 (SC7) / Working Group 6 (WG6).

software engineering has not allowed full standardization. Subsequent improvements or developments could have substantially delayed the publication of ISO / IEC 9126. Moreover, it has been considered that a continuation of research in many countries could lead to and different solutions whose harmonization would then be costly.

Based on these considerations, SC7 / WG6 published in 1991 a version of the international standard ISO 9126 to harmonize further developments. This standard was adapted as a Romanian standard.

Since 1991, SC7 / WG6 has started to develop standards that deal in detail with the software evaluation process. The activities have been finalized by developing and publishing ISO Standards Series 14598 during 1998-2001.

The publication of Series 14598 has led to the major revision of ISO 9126 in the following main aspects:

- transferring the evaluation process to the ISO standards Series 14598;
- detailing the quality features and sub-features;
- modifying definitions of quality features and sub-features, as well as concepts related to measurement (metrics and measures);
- clarifying the relations between external quality, internal quality and quality in use;
- identifying and defining a set of metrics grouped into three categories (external metrics, internal metrics and metrics of quality in use).

As a result of these revisions, the ISO 9126 version of 1991 was replaced by four standards that form Series 9126.

Thus, the domain of software product quality is currently covered by two sets of standards (see Table 1).

Except for parts 2, 3 and 4 of Series 9126 that are published as “Technical Report” (TR), all other standards are published as International Standards (IS). The reason for publishing as TR documents is because measurement methods and quality metrics have not yet reached maturity and recognition in the international community.

Table 1 List of ISO standards related to the quality of software products (31.08.2004)

ISO/IEC 9126-1:2001	Software product quality. Part 1: Quality Model.
ISO/IEC TR 9126-2:2003	Software product quality. Part 2: External Metrics.
ISO/IEC TR 9126-3:2003	Software product quality. Part 3: Internal Metrics.
ISO/IEC TR 9126-4:2004	Software product quality. Part 4: Quality in Use Metrics,
ISO/IEC 14598-1:1999	Software Product Evaluation. Part 1 : General Overview.

ISO/IEC 14598-2:2000	Software Product Evaluation. Part 2: Planning and Management.
ISO/IEC 14598-3:2000	Software Product Evaluation. Part 3: Process for Developers.
ISO/IEC 14598-4:1999	Software Product Evaluation. Part 4: Process for Acquirers.
ISO/IEC 14598-5:1998	Software Product Evaluation. Part 5: Process for Evaluators,
ISO/IEC 14598-6:2001	Software Product Evaluation. Part 6: Documentation of Evaluation Modules.

Figure 4 shows the relationships between the two sets of standards.

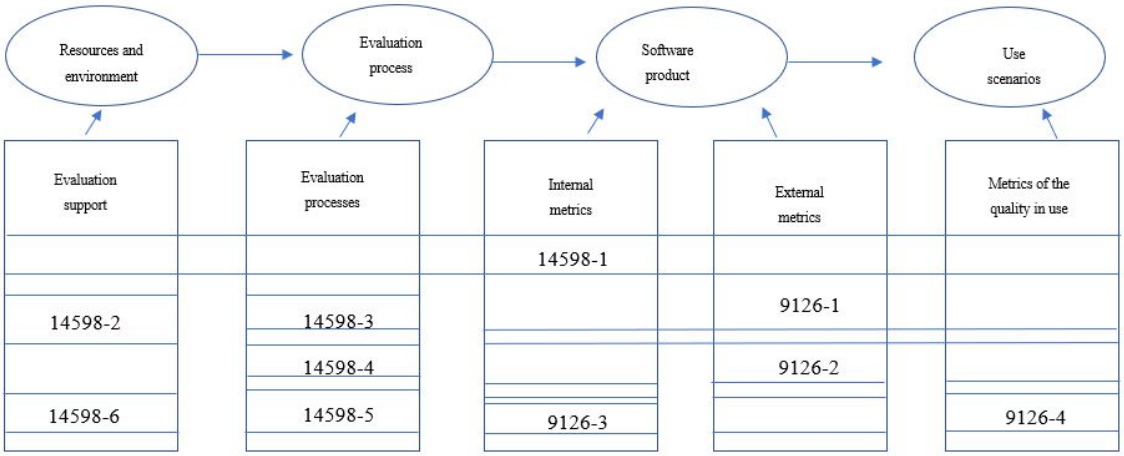


Figure 4 Relationship between ISO Standards Series 9126 and 14598 (source: adapted from ISO 14598)

Further, the object and the purpose of each part of ISO Series 9126 and ISO Series 14598 are presented.

ISO 9126-1 describes the software product quality model that includes internal quality, external quality and quality in use. The standard defines six features of internal and external quality, which are further broke down into sub-features. The standard also has four features of quality in use. It is used for the following main purposes;

- specifying quality requirements;
- evaluating product quality;
- identifying quality assurance criteria and acceptance criteria for a software product.

ISO 9126-2 describes the set of external metrics and recommendations on how to use them to measure a feature, sub-feature, or external attribute of the software product. The metrics presented in this part of the standard are used for the following purposes:

- specifying quality requirements;
- evaluating product quality in the final test and acceptance phase;
- researching, experimenting, and developing new metrics.

ISO 9126-3 describes the set of internal metrics and recommendations on how to use them to measure an internal attribute of the software. Metrics are used for the following purposes:

- defining the objectives of the project;
- analyzing the quality of intermediate products (specifications, source code, etc.);
- researching, experimenting, and developing new metrics.

ISO 9126-4 describes the set of quality in use metrics and recommendations on how to apply them to measure the effects of using a software product. Metrics are used for the following purposes:

- defining the objectives of the project;
- defining the acceptance and evaluation criteria for final products.

ISO 14598-1 includes the general requirements and recommendations for specifying and evaluating the quality of software products, and definitions of the terms used in other parts of the standard. It also provides the general framework for evaluating the quality of all types of software products and defines the requirements for measurement and evaluation methods of software products.

The standard applies for the following purposes:

- planning the evaluation of software products;
- implementing the evaluation process;
- carrying out the evaluation process;
- measuring the product attributes;
- performing the evaluation;
- assessing the evaluation process in relation to different objectives (e.g., determining the delivery time of the product, making the product acceptance decision, comparing the product with other similar products, selecting a product among several available products).

ISO 14598-2 provides a set of requirements and recommendations for the management of support processes for software product evaluation. It also includes recommendations for developing and using a measurement plan. It is used together with the ISO/IEC 9001:2000 and ISO/IEC 90003:2004 standards.

ISO 14598-3 provides requirements and recommendations for evaluating the software product during the design and development processes:

- criteria for selecting metrics;
- recommendations for verifying and validating quality features;
- recommendations for measurement analysis;
- recommendations for improving the evaluation process.

This part of the standard is used for the following purposes:

- identifying quality requirements;
- monitoring and quality control during product development;
- validating intermediate and final products;
- identifying the changes needed to meet the requirements.

ISO 14598-4 provides requirements and recommendations for evaluation in order to select and accept a product. This part establishes a systematic software evaluation process in order to decide on product acceptance or selection of a product from several others based on the ISO/IEC 9126 quality features and metrics and based on the software product evaluation model as described in the ISO/IEC 14598-1:1999 standard.

It allows the evaluation process to be adapted to achieve a level of trust in the software product, consistent with the nature and level of integrity of the application.

ISO 14598-5 provides a set of requirements and recommendations for the implementation of the evaluation process. It defines the activities necessary for:

- analyzing the evaluation requirements;
- specifying the evaluation project;
- carrying out the evaluation;
- defining the conclusions of the evaluation.

It is used to evaluate software deliverables as well as, when required, to plan and implement the evaluation process, or analyze the results of the evaluation.

ISO 14598-6 defines a set of requirements and recommendations on the structure and content of the documentation used to describe an “evaluation module”[2]. It also contains recommendations on developing and validating evaluation modules.

At international level, the two sets of standards have been adapted and implemented in many companies (e.g., IBM, HP, Motorola, Ericsson, Mitre Corp., DELTA Electronics, Raytheon, Schlumberger, Wipro and so on) and in software testing laboratories.

[2] “Evaluation module” is a concept used in ISO 14598 designating a coherent set of evaluation methods and procedures, pre-defined reports, tools, and so on, used to measure and evaluate the features, sub-features and attributes of the software product quality.

Internally, the series of standards in the various published versions were used in the period 1993-2000 within the ICI (National Institute for Research and Development in Informatics) to develop a methodology for testing and evaluating the quality of software products for certification purposes. At the same time, the standards have been applied in experiments aimed at evaluating the quality of software products for public administration and evaluating the quality of multimedia products.

1.2.2 The New Generation of ISO Standards Series 25000

Between 2001 and 2002, ISO/IEC SC7 WG6 members undertook a series of actions to verify the applicability and relevance of the 9126 and 14598 series of standards. An investigation was carried out to collect information from many sources (e.g., information technology firms, research teams from universities, independent consultants, etc.).

WG6 experts completed the actions in a report that highlighted the need to improve software product quality standards in the following directions:

- ensuring the completeness of standards (e.g., including recommendations on specifying quality requirements and procedures for checking metrics);
- ensuring the consistency of standards with other related standards (e.g., alignment of terminology to ISO standards of terminology, compliance with measurement standards, etc.);
- clarifying the scope of application (e.g., including clarifications on the definition of quality needs in the life cycle of the product, including additional methodologies or guidance for the application of standards, etc.).

The actions taken led to two strategic decisions for ISO SC7/WG6:

- finishing and publishing all standards in the 9126 and 14598 series by 2003;
- re-analyzing, prioritizing and considering all identified improvements, and initiating the development of the second generation of software product quality standards: **ISO/IEC 25000 SQuaRE** (Software Product ***Qua***lity ***R***equirements and ***E***valuation).

The main objective of the new ISO/IEC 25000 series is to respond to the evolving needs of users through a set of improved and uniform regulatory documents covering three complementary processes: specifying quality requirements, quality measurement and quality evaluation (ISO/IEC FCD 25000, 2004).

The most important improvements envisaged for the next generation of standards refer to the following:

- including the requirements of the previous standards series 9126 and 14598 into a single harmonized structure;
- introducing a new organization of standards;
- introducing a new quality measurement model;

- introducing detailed recommendations and guidance;
- introducing a standard on elementary measurement attributes;
- introducing a standard for determining quality requirements;
- introducing guidance, including examples, on the application of the set of standards;
- harmonizing the software product quality model with the reference model in ISO/IEC 15939:2002;
- establishing and clarifying relationships with other process standards (e.g., ISO/IEC 15288:2002, ISO/IEC 15504, SR ISO CEI 12207:2000), including the ISO 9001:2000 standards.

The new generation of ISO/IEC 25000 standards will include 14 normative documents grouped into five thematic groups (see Figure 5):

- **Quality management.** The standards in this thematic group have the SQuaRE architecture, they define the terminology and conceptual models referenced by the other parts of the SQuaRE series. It also provides the requirements and guidance needed to manage the specification of requirements, measurement and evaluation of software products.
- **Quality model.** The standards in this thematic group have two parts of the general quality model (the external and internal quality model, the quality in use model), defines quality features and sub-features.
- **Quality measurement.** The standards included in this thematic group describe the software product quality measurement model derived from the reference model specified in ISO/IEC 15939:2002. It also describes a set of measures (over 100) that can be used to measure external quality, internal quality and quality in use, as well as the elementary measures (measurement primitives) used to obtain the other measures.
- **Quality requirements.** The standards that form this thematic group help to specify software product quality requirements. The process of defining and specifying requirements is harmonized with the technical processes defined in ISO/IEC 15288:2002.
- **Quality assessment.** The standards that form this thematic group define the requirements and describe the recommendations for evaluating the software from three perspectives (developer, purchaser and evaluator). In addition, the evaluation process considers the requirements of SR ISO 12119:1999 and is harmonized with the technical processes defined in ISO/IEC 15288:2002.

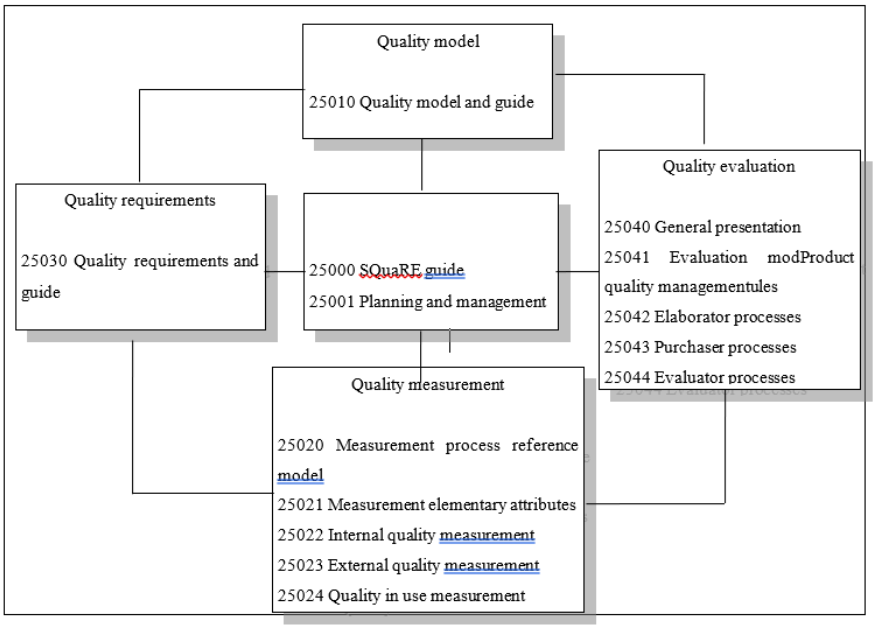


Figure 5 Parts of ISO Series 25000

SQuaRE series numbering was approved at the ISO JTC1/SC7 plenary meeting in Busan, South Korea, in May 2002.

The new set of standards will include both new standards and improved standards through the revision of the 9126 and 14598 series of standards (see Table 2):

- 4 new standards, some of which are developed by reviewing and unifying the relevant common parts of the previous standards 9126 and 14598;
- 5 improved standards through the major revision of previous standards (especially the 9126 series);
- 5 improved standards through the minor revision of the 14598 series standards.

Table 2 Correspondence between the parts of ISO 25000 and ISO 9126 and 14598

Series 25000	9126-1	9126-2	9126-3	9126-4	14598-1	14598-2	14696-3	14598-4	14698-5	14596-6	Obs.
25000											new
25001											min
25010											Maj

25020											new
25021											new
25022											Maj
25023											Maj
25024											Maj
25031											new
25040											Maj
25041											min
25042											min
25043											min
25044											min

Legend: *new*: new standard

min: minor revision of previous standards

Maj: major revision of previous standards

2.3.1.2 Other Standards

In parallel with the development of software product quality standards, other ISO working groups have developed standards that support and / or provide details on the understanding and application of the different requirements of the 9126 and 14598 series (see Table 3).

Table 3 Other standards supporting the application of the 9126 and 14598 series

ISO 9241-11:1994	Information Technology - Ergonomie requirements for office work with visual display terminal (VDTs) - Guidance on usability.
ISO/IEC 12119:1994	Software Engineering - Software packages. Quality requirements and testing.
EN ISO 13407:1999	Human-centred design processess for interactive systems.

ISO/IEC 14143-1:1998	Information technology - Software measurement - Functional size measurement - Part 1 : Definition of concepts
ISO/IEC 14143-2:2002	Information technology - Software measurement - Functional size measurement - Part 2: Conformity evaluation of software size measurement methods to ISO/IEC 14143-1:1998
ISO/IEC 14143-3:2003	Information technology - Software measurement - Functional size measurement - Part 3 : Verification of functional size measurement methods
ISO/IEC 14143-4:2002	Information technology - Software measurement - Functional size measurement - Part 4: Reference model
ISO/IEC 14756:1999	Information technology - Measurement and rating of performance of computer-based software systems
ISO/IEC 14915-1:2002	Software ergonomics for multimedia user interfaces - Part 1: Design principles and framework
ISO/IEC 14915-2:2003	Software ergonomics for multimedia user interfaces - Part 2: Multimedia navigation and control
ISO/IEC 14915-3:2002	Software ergonomics for multimedia user interfaces - Part 3: Media selection and combination
ISO/IEC 15026:1998	Software Engineering - System and Software Integrity Levels.
ISO/IEC 15939:2002	Software Engineering - Software Measurement Process Framework.
ISO/IEC 19761:2003	Software engineering - COSMIC-FFP - A functional size measurement method
ISO/IEC 20926:2003	Software engineering - IFPUG 4.1 Unadjusted functional size measurement method - Counting practices manual
ISO/IEC 20968:2002	Software engineering - Mk II Function Point Analysis - Counting Practices Manual

2.3.2. Product quality in the life cycle

2.3.2.1 Software characteristics

The quality terms reviewed in Chapter 2.1 are too general to be applicable in a narrower sense to software. The definition of *software* is not enough to cover everything that is not *hardware*. As defined by the IEEE Computer Society (IEEE Std 610.12:1990), software includes all programs, processes, documentation and data related to the operation of a computer system.

Based on this, software is perceived as a technical product that doesn't need to take a special role in relation to the functionality and quality of all other products. "Whoever emphasizes the particularity of software asks an unfounded privilege" (Jochen & Lichter, 2010, p.34). However, software has features that have an impact on quality assessment and will therefore be examined below. Jochen & Lichter (2010, p.34) show the essential features summarized below.

Immateriality

Software is delivered and used through real objects as well. Usually, when talking about software, these are not of interest. Software generally divides the qualities of computer science, namely its essential parts, and the actual significance remains abstract (Coy, 1989, p. 256-266).

Software is developed

Jochen & Lichter refer to the fact no relevant production costs are generated for software, if production means the reproduction of the first unit developed.

Software does not wear out

On the one hand, reproduction does not cause losses, as the copy corresponds to the original, on the other, software does not lose this feature through use. For this reason, software reuse is not economically attractive, if no adjustments are needed. Compared to the maintenance of material goods, maintenance does not represent the starting state but puts the system into a new state deviating from the original. Also, software errors are not generated by usage but by changes or the occurrence of inherent errors.

Natural location

In the real world, distance is related to the fact that the parts of a system influence each other less the greater the distance between each other (Jochen & Lichter, 2010, p. 35). The fact that software is not covered by the laws of nature and that the most remote parts may affect each other means that it must be separated by appropriate rules and artificial structures, such as modularization and hierarchy.

Software does not perform a permanent function

Another special feature of software is that the program does not perform a permanent function (Jochen & Lichter, 2010, p.35). This means that the link between the input and the output cannot be described by a function. Therefore, a change of a single bit on the input side can attract unforeseen changes on the output side (Coy 1989, p. 256-266).

System complexity

Balzert (1998, p.6) refers to various dimensions of complexity, presented by six classes of complexity: functions, data, algorithms, behavior depending on time, system environment and user surface complexity.

2.3.2.2 Goals and principles in quality assurance

Software testing is a process for verification and validation a software product. This approach serves to ensure that the initial requirements for the product have been met and to recognize errors and deficiencies to be remedied for the product to perform as expected.

The process of testing software is one area of software quality assurance that can recognize only 80% of errors and deficiencies in large software products by using more resources, but this process is not limited to performing tests. (Capers & Bonsignour, The Economics of Software Quality, 2011, p.344)

Goals that are pursued in the process of testing software are, finding errors, collecting information for new decision or preventing future errors. Both activities before and after the tests need to be included. These “additional” activities include, for example, test planning and control and reporting after the test. (Müller, et al., 2011, p. 13)

The performance of the test process is based on some test goals and test principles which are explained in the following. The goals define the desired result of the process and the principles describe the elementary models that can be used for effective and efficient testing. (Quardi & Farooq, 2010, p. 1-2)

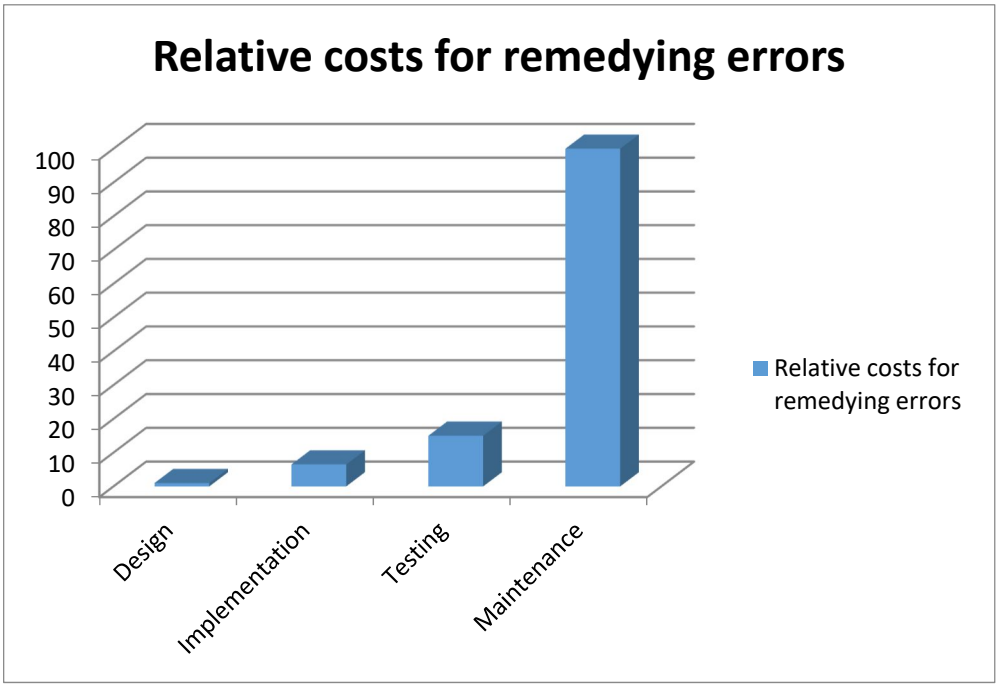


Figure 6 Relative costs for remedying errors

It should be considered that in accordance with the actual level, this cost comparison is far from being sufficient to measure software quality. Thus, according to Capers & Bonsignor, the utility of software quality is made in total of seven points:

1. Reduced likelihood of system crashes
2. Reduced likelihood of legal disputes for external developments
3. Reducing the development period
4. Reducing development costs
5. Reducing maintenance costs
6. Reducing warranty costs
7. High customer satisfaction

Testing control the presence of errors

Testing itself does not have any effect on the quality of a software product. The central task of the test is the discovery and documentation of errors. Any other task, such as remedying the problem, lies outside the scope of the testing process. (Meyer, 2008, p. 1)

It is not possible to test everything

As mentioned briefly in the introduction, it is not helpful and lost time to test the software 100%. According to current statistics, this is not necessary. Therefore, even software

errors follow the Pareto principle according to which 80% of software errors which are discovered during the testing process can be found in 20% of software modules.

It's worth it to test in advance

The costs of remedying errors are important in order to remove them as early as possible because remediation costs in later development phases increase by about 100 times.

(Müller, 2011, p. 14)

Error grouping

Tests should be performed systematically for few critical modules; in practice it has been shown that a small number of modules contains much of the errors because the probability of other errors behaves according to the number of errors already found. (Müller, 2011, p.14)

The paradox of pesticides

By continuously repeating tests there is the risk that these texts can no longer recognize new errors, therefore test cases should be checked and updated regularly. (Müller, 2011, p.14)

Testing depends on context

Tests must be adapted to context to achieve success. Therefore, safety-critical applications must be tested differently than performance-critical applications. (Müller, 2011, p.14)

Falsity of the lack of errors

An error-free application may not necessarily have to be a useful application. It must also be ensured that applications continue to meet customer requirements. (Müller, 2011, p.14)

2.3.2.3 The structure of the testing process

After so far, we described some reasons, goals and principles of the testing process, the next subchapter deals with the structure and the conduct of the testing processes. The basis to describe the testing process is the definition of *International Software Testing Qualifications Board* (ISTQB), which can be found in Foundation Level Syllabus by Certified Tester Program (Müller, 2011). ISTQB defines the testing process by the five steps of test planning and control, test analysis and development, test achievement, assessment of final criteria and the completion of the testing.

Test planning and coordination

Test planning and coordination are regular activities during the entire process of testing, where test planning also represents the starting point of the process.

The purpose of test planning is represented by all the testing goals and activities needed to achieve the overall goal of the project. In this case, test planning have to be reconciled with the overall planning and must consider both the testing strategy, existing resources, responsibilities, and the risks and priorities.

Outside planning, test coordination ensures the current monitoring of the initial testing plan to recognize possible deviations from the plan. Additionally, it represents the task of test coordinating to take the appropriate emergency measures so that the original test plan can still be achieved.

A part of the master test plan describes the so-called integrity levels. They define how important are the individual parts of the software product to the user, based on IEEE specifications, with reference to all areas of the testing process. They can be used for requirements, test specifications or components. The basic principle for the division of levels of integrity is always governed by different characteristic sizes, such as complexity, risk, level of safety, performance or reliability. (IEEE Computer Society, 2008, p.13)

A list of possible levels of integrity, which were taken from the IEEE standard, can be found in Table 4. Based on such a division, in the next phases of testing decisions is taken on the effort and priority of individual test cases.

Table 4 Level of integrity according to IEEE829-2Q08 (IEEE Computer Society, 2008, p. 13)

Level of integrity	Description
1	Software errors have very serious consequences (e.g., system crash, death, environmental damage). It is not possible to mitigate damages in the event of an error.
2	Software errors have serious consequences (accidents, environmental pollution, social or economic losses). The decrease of damage is partially possible.
3	Software errors have reduced consequences. It is possible to mitigate damages in full.

4	Software errors have consequences that can be neglected. It is not necessary to diminish the damage.

Analysis and development of the test

General testing goals are set while planning; it is the task of analysis and test development to describe as accurately as possible test cases with all their prerequisites and requirements. Specifications in the planning phase are used as a basis for this, checked and expanded accordingly. Thus, the development phase should consider not only test data and conducting the test. Additionally, it must be checked which basic premises must be met for the particular test cases. This includes, for example, additional tools, training and simple equipment. The result of this phase is represented by the so-called test cases.

Test performance

As apparent from the name, tests are performed in this phase relying only on their planning and configuration. Additionally, this phase should also consider the testing environment and the order defined during the configuration phase. The results of the performance phase are test and error reports and status and completion messages.

Assessment of final criteria and the report

The goal of assessment phases is the control of the final test results. In this process, test end reports are compared with the test goals from planning and design phase. This can evaluate whether the testing process has been completed or if further tests are needed. This procedure generates a circuit to be covered until all the necessary conditions are met.

A total report for all stakeholders is drafted at the end of the assessment phase. (An example can be seen in appendix B)

Completion of test activities

Completion of the testing process is used to gain experience. This phase is usually carried out in the phases of the project, such as software release, completion of a testing project or a new maintenance cycle. This phase retains the successful, failed or challenging events that will be used to further improve processes.

2.3.2.4 Bases of measurement and software quality

Bases of measurement

As shown by Kan in (Kan, 2003, p.55), measurement is a fundamental contribution to science development. In principle, scientific progress is achieved through observation and generalization based on data and measurements. Theories are elaborated and checked based on empirical data or even falsified. We presented as an example the theory that argues that implementing a consistent software development process in the early stages will lead to high quality results at the end. To prove the correctness of this theory, firstly we must define all the existing individual concepts meaning *early stage* in this context.

First, we must define the development of software as a concept related to measurements as described in (Kan, 2003, p.56):

- Design
- Design reviews and verification
- Implementation
- Verification of the implementation
- Remediation of mistakes and development tests
- Integration at components and module level
- Formal testing on the fulfillment of requirements
- Input phase at the customer

After the integration phase, the software product is subject to change-management, which means that changes to the software are made more on the basis of special requirements (e.g., errors that were not discovered by development tests). So this must be defined as the separation between the early stages and late phases of a software project. The phases are implemented strictly in accordance with guidelines specified in the process. For assessment, if the process was conducted according to guidelines, the assessment criteria and the necessary data on which the theory can be described as false or qualified are missing. These criteria or indicators serve to operationalize the assessment. Examples for these indicators are Lines of Code (LOC), which are subject to scrutiny in

the implementation phase. Another metric can be the effectiveness of verification. Lickert[3] proposes a scale that is composed of five values.

- 5: highly effective
- 4: effective
- 3: almost effective
- 2: ineffective
- 1: absolutely bad verification

Operational assessment criteria must be defined even for all the early stages, such as the number of instructions covered by tests or the number of errors per 1,000 lines of code that were remediated. To make an instruction later if the desired quality of the result was achieved the late stages of the development process of the software must also be provided with operational criteria. Here we may give as example the metric *The number of errors found pro KLOC (1000 source code lines)* during formal testing. Based on metrics, a hypothesis can be developed that states that for the software project (Kan 2003, p.57)

- the error rate in late stages is the lower the higher the percentage rate of the artifacts checked in early stages,
- the more effective the checks in the early stages, the lower the number of errors in late stages and
- the more intensive the tests before the integration phase, the lower the number of errors found during formal testing.
- According to this hypothesis, necessary data can be collected, and the hypothesis can be verified or invalidated. This requires the determination of the artifacts to be analyzed (classes, methods …). With a sufficient number of points to be measured on representative number of samples, the information required can be found by statistical procedures.

2.3.2.5 Levels of measurement (types of scales)

As seen from the introduction to this chapter, a definition must be operationalized for the subsequent assessment (in this case, the development process of the software) and indicators for the following measurements must be deducted. The scales of those indicators should also be considered. In the example shown above, to measure the quality of a software check we used a scale of five positions for the assessment of the effectiveness of the check. In some cases, multiple scales can be used, while in other

[1]http://www.wirtschaftslexikon24.net/d/likert-skala/likert-skala.htm

cases the nature of the basic concept provides a certain scale or measurement level. In the following subchapters we discussed the four levels of measurement: nominal scale, ordinal scale, interval scale and relation scale (Kan 2003, p.59).

Nominal scale

The simplest type of scale is the nominal scale which represents a simple classification based on the attributes of an object. As an example, software architectures can be divided into one-, two- or three-layer architectures. All these architectures with more than three layers are generally known as multilayer architectures. The premises for implementing a nominal scale are that they integrated together, namely are jointly exhaustive and mutually exclusive. Another characteristic of the nominal scale is the fact that it has no inherent order. Thus it cannot be argued that a one-layer architecture is better than a three-layer architecture.

Ordinal scale

In software development, projects can be classified on the observance of a software development process (*process rigor scale*), total compliance (*totally adhere to process*), partial compliance (*somewhat adhere to process*) or noncompliance *(does not adhere to process)*.

The ordinal scale ranges relative to the measuring process over the nominal scale. Thus, in addition to a grouping of categories, an *ordering* is also possible. The ordinal scale is transitive and in it is true that if $A > B$ and $B > C$ then $A > C$. This is asymmetric, meaning that if $A > B$ is true, then $B > A$ is false.

However, the ordinal scale does not indicate anything about the difference of sizes between two categories. In relation to *process rigor scale* this means that we know only that *totally adhere to process* is better related to the quality of the software than *somewhat adhere to process*.

Interval scale

An interval scale gives a precise difference between measurement points. It follows that mathematical operations like addition and subtraction can be used on these interval data. For example, considering that three software products A (8 defects per KLOC), B (4

defects per KLOC) and C (2 defects per KLOC) have been developed with the same software language, the following statements can be made about their rate of failure:

- the fault rate in A is 4 faults higher per KLOC than in B
- the difference of fault rates between A and B is twice as high than between B and C

Relation scale

The relation scale is an expansion of the interval scale if the scale does not possess an absolute random point O. The relation scale is the highest scale that can be applied to all mathematical operations. To apply the example above with fault rates, it can be stated that the fault rate in A is twice as high as in B. This is given by the fact that a fault rate of 0 means that there is no malfunction.

Relation scales will be used for all practical purposes, because it is almost always possible to determine the point 0 of a scale.

Finally, it should be noted that these scales are hierarchical. This means that any superior scale inherits all the qualities of the inferior scale. It can be stated that assessments through superior scales allow better statements about the systems analyzed.

Basic sizes

Irrespective of measurement scales, accumulated values need to be analyzed in order to obtain adequate information from them. For this, different sizes and statistics are available.

Relation

The coefficient between two numbers is called relation. It should be noted that the numerator and the denominator must come from two different populations which shall be mutually exclusive.

$$\frac{\text{Number of independent testers}}{\text{Number of developers}}$$

The above formula shows the relation between specialized test engineers in relation to the developers of a company. The smaller the relation, the more test work must be carried out by developers.

Ratio

In ratios, the numerator is a component of the denominator.

The relative frequency is when the numerator and the *denominator* are integers, meaning that they relate to counting events. Thus we may indicate the ratio of satisfied customers to report the total number of customer as

$$\frac{\text{Number of satisfied customer}}{\text{Total number of customers of a software product}}$$

If the numerator and the denominator are not integers, then these ratios are called fractions.

Percentage

A percentage is a relation or a fraction of a *hundred* units of calculation. This means that the denominator is normalized to 100. Percentages are often used to represent results. Because percentages represent relative frequencies, the value generation context must be indicated with the values. Therefore, the total number of analyzed events must be indicated to give the person performing the analysis the opportunity to estimate the significance of values of numbers.

2.3.2.6 Metrics of software quality

Like software quality, software metrics can also be subdivided into different categories (Kan, 2003, p.85). The following categories may be distinguished:

Product metrics

Characteristic of a product as size, complexity, design, performance.

Process metrics

Characteristics of a software development and maintenance process as the effectiveness of error remedy during development or the duration until the error is remedied since it was notified.

Project metrics

Project and achievement characteristics, like the number of developers involved or the number and the respective roles of project members throughout the project (life cycle of the software).

Software quality metrics are a subset of software metrics that assess product, process and project quality. Among these, product and process quality are generally considered more accurately. Therefore, software quality metrics are subdivided into *Final product quality metrics* and *Process quality metrics*. The challenge to establish software quality is finding correlations that determine the quality of process metrics, project characteristics and product quality, thus subjecting both process quality and product quality to continuous improvement.

This subchapter will analyze product quality metrics. It is crucially important to increase the power of expression of metrics to the person that performs the analysis and its visualization possibilities.

Product quality metrics

Mean Time To Failure (MTTF)

The interior quality of the software is often determined by the number of errors contained or how long it takes to a *crash*.

Mean Time to Failure (MTTF) is mainly used in safety critical applications such as those used in aviation and weapons systems. This involves establishing limit values, such as the system cannot be unavailable for more than three seconds per year. The practical value of this metric for standard software assessment is rather low. First, collecting the necessary values involves a great effort. On the other hand, values depend on the run times and categories of users exactly defined. These conditions are more easily met for special software than for standard software.

Error density

Error density is defined as (Defect Density Rate) the number of defects per opportunities for errors (Opportunities for Errors) in a certain time interval. The number of defects can

be determined as a random error behavior of a system due to errors found at its base. This can prevent that an error that is responsible for a different erroneous behavior to be counted several times. This number will be divided by the theoretical possibilities of error. As a theoretical opprotunity for error, the number of lines of source code (lines of code) of the artifact to be analyzed can be used. This rate will be normalized to 1000 lines of source codes.

$$\frac{\text{Number of errors}}{\text{Number of source code lines}} * 100$$

As time frame we can set a period between one to four years after the implementation of a software system. In standard applications, only 95% of the errors are typically found in the first two years of implementation.

Lines of Code (LOC)

The Lines of Code metric seems very simple at first glance. However, some aspects have to be considered so that the information obtained be interpreted correctly and no wrong conclusions be drawn.

The most important issue is the counting. This is motivated already in the origin of this metrics, which related to the assembler code. In the assembler code, every line of source code should be viewed as an instruction. This simplifies the counting process and allows the interpretation of the counting values obtained. With the advent of standard languages, this direct link between instruction and the source code line was lost. In addition to this problem we must highlight the fact that different standard languages cannot be directly compared. This is mainly due to the different amplitudes of standard languages. For this reason, the norming of standard languages on LOC per assembler codes was attempted. But this was not performed for all standard languages.

In addition to the above issue, it is difficult to determine what can be counted as a line of source code. Kan (Kan 2003, p.88) shows the following counting variations:

- Lines of only executable codes
- Lines of executable codes and definitions of variables
- Lines of executable codes, definitions of variables and comments
- Lines as physical lines on the screen
- Lines as logical lines completed by logic limiters (for instance)

Additionally, structures of modern languages, such as *annotations*, should also be introduced in the above analysis.

Differences in the values of numbers obtained depending on the counting version used can lie in a range up to 500%. Thus, it is important to indicate the context of counting (language, the definition of a line of source code, etc.) for any indication of values that relate directly or indirectly to this metric to give the reader the opportunity to compare and adapt analyses. The comparison by different programming languages is possible only through assembler equivalents. Without assembler equivalents, comparisons are not possible on the density of errors through programming languages.

Cyclomatic complexity

The term cyclomatic complexity was first used in 1976 by McCabe. It serves as a measure for testing and the perception of programs codes. This amount is deducted from the cyclomatic number known in the classical theory of graphics, indicating the number of independent branches in a graph. In graphics development, it relates to all independent linear flow control channels occurring in a program. From this, the upper limit of the testing procedures can be determined to obtain a complete coverage test of the branch of a program.

The general formula for cyclomatic complexity is:

$$M = V(G) = e - n + 2p$$

in which

- V(G) = the cyclomatic number of graph *G*
- e = number of edges
- n = number of nodes
- p = number of analyzed graphs

Figure 7 shows a simple graph as example

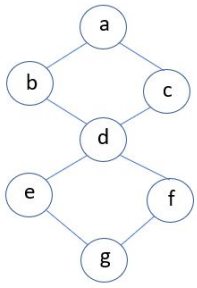


Figure 7 Control graph according to [Kan03, p. 316]

This graph shows a simple program with two branches (e.g., IF-Statements). Counting individual elements gives the number of edges *e* with 8, of nodes *n* with 7 and the number of graphs analyzed *p* with 1. Hence the cyclomatic number

M = 8 - 7 + 2 * 1 is calculated with the value 3. As a rule, it can be established that *M,* always the number of all binary decisions (IF, ITERATION) increased by 1, is a three-way decision and will be counted as two binary decisions. A new path *ncase*-Statement is treated as *n-1* binary decisions.

Cyclomatic complexity is additive, which means that full complexity of multiple graphics analyzed is equal to the sum of individual complexities. It should however be noted that this metric does not analyze sequential complexities. Even the inherent complexity differences of individual flow control statements (IF-THEN-ELSE vs. loops or IF-THEN-ELSE Statements complicated vs. SWITCH) are not taken into account by this metric.

McCabe indicates the value 10 for the upper limit of the cyclomatic complexity for a software module. When exceeding this upper limit, the module should be divided. This is a very simplified rule because cyclomatic complexity analyzes only control flow graphs. Other complexity influences, such as data flow complexity or the different complexity of different control flow statements, are not examined. The decision to split a module must always be achieved by observing several metrics.

Structure metrics

Structure metrics analyze the interaction of several software modules with each other and try to quantify this interaction. The best known are the *fan-in* and *fan-out* metrics.

- *fan-in:* Indicates the number of modules selecting a module analyzed
- *fan-out:* Indicates the number of modules selecting a module analyzed

Generally, modules with a high *fan-in* are rather small and simply structured, which includes a high degree of reuse. These modules are often also the basic components of a software. Compared with these, large, complex modules have only small *fan-in* values. As noted by Kan (Kan 2003, p.320), modules with a high *fan-in* and *fan-out* have a bad configuration and should be subject to reconfiguration.

Flenry and Kafura defined structural complexity as (Kan 2003, p.320):

$$C_p = (fan - in \times fan - out)^2$$

This idea is continued by Card and Glass who define system complexity as:

$$C_t = S_t + D_t$$

in which C_f is the complexity of the system, S the structural complexity and D_f the complexity of data. The peculiarity of this extension is to involve the complexity of data under the assumption that the number of variables I/O of a module directly influence its complexity. This occurs because the I/O variables in a module are needed for the performance of the functionality. The structural complexity *S* is defined by Card and Glass as

$$S = \frac{\sum f^2(i)}{n}$$

In which $f^2(i)$ is the *fan-out* of module number *i*, and *n* is the number of modules. The cubic dependence of the structural complexity of a module is based on knowledge of different analyses. It was found that the *fan-in* number contributes less to structural complexity, but unlike this the complexity performs a cubic increase with the *fan-out* number.

The complexity of data D of a module *i* is defined as

$$D_i = \frac{V(i)}{f(i)+1}$$

V(i) designates the number of variables I/O and f (i) *fan-out*-number of module *i.*

The complexity of data is directly dependent on the number of variables I/O but indirectly by the *fan-out* number. This dependence indirect of the *fan-out* number is justified by delegating functionalities on other modules by selecting them.

The whole complexity of the data is defined as the average complexity of data on all modules

$$D = \frac{\sum D(i)}{n}$$

D(i) designates the complexity of the data of module *I* and n the number of modules.

Project metrics and limit values

If metrics must be used, it must be established in advance what results of measurements are *too high* or *too low* and what *too much* or *too little* means. This means that we need to set the limit values of corresponding metrics and establish the so-called benchmarks that determine the semantics of certain metrics. This allows the accurate interpretation of individual values measured. The limit values share the domain of possible values of results in individual regions. Depending on the region to which a value of the measurement is assigned, an established assessment can be performed.

There are several testing metrics which can be used to measure the performance of testing process such as defect detection percentage. Defect detection percentage measures the overall quality of company's testing process. It is the ratio of a number of defects identified during testing divided by total defects identified in that phase. Greater defect detection percentage indicates a reliable and effective testing process. It also increases chances that you deliver a bug-free product to your client.

DDP = (Number of defects detected in a phase / Total number of defects in that phase) x 100 %

For example, if 75 defects were found in testing and 25 additional flaws were found in the two months following launch, you have a Defect Detection Percentage of 75%. It means that you caught only 75% bugs during testing.

Once these metrics were understood, it is time to apply the knowledge and analyze the testing methods. For this purpose, can be used a defect tracking system, that add details of 'detection phase' whenever a bug is logged. In this way, it will be able to identify how successful you have been in preventing defect leakage.

Calculate 'Defect detection percentage' for every testing cycle to get a measure of detecting the bugs. This percentage can to calculated on the basis of different QA Teams. This will give an insight of which team is performing up to the mark. It would also enable you to identify flaws in testing process.

Object-oriented metrics

The above metrics can be used starting from software development in assembler to the development with imperative languages such as C or Fortran. Using the object-oriented programming paradigm, these metrics should be changed to ensure in this paradigm also the desired power of expression.

The aim of the object-oriented paradigm is to simplify development, and in particular, to maintain the software. The starting point for the success of this paradigm has been the much-quoted software crisis that began in the mid '60s and peaked in the early '70s. The term software crisis was first mentioned by Edsger W. Dijkstra in 1972 in his speech (The humble programmer at the Turing award ceremony). In it, he explained, the major cause [of the software crisis] is …that the machines have become several Orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak Computers, programming became a mild problem, and now we have gigantic Computers, programming has become an equally

gigantic problem. In this sense the electronic industry has not solved a single problem, it has only created them - it has created the problem of using its products.

The fact that the size and therefore the complexity of machines (software) is the biggest problem to solve was recognized since then. The object-oriented programming paradigm intervenes here. On the one hand, an overview of the generally valid functionalities must be ensured to the specific characteristic by the hierarchical order of abstract data types (abstract classes) through a deduction hierarchy. On the other hand, it prevents by *Information hiding* the uncontrolled access on the state of the program and the state of an object. A circumstance that cannot be prevented in imperative languages leads to many problems in running programs and still does (C ++). Proper use of concepts of the object-oriented programming paradigm discussed here have to reduce the complexity of a component and of an entire program at a level of complexity that can be handled.

Most metrics mentioned in this paper will act directly or indirectly, in particular on the complexity of an artifact to be analyzed of an object-oriented program (components, classes, methods). The configuration of components is another criterion whose quality can be measured by object-oriented metrics and thus can be assessed operationally.

Mark Lorenz has already proposed in 1993 eleven object-oriented metrics. For these metrics he provided *Rules of Thumbs* and comments on how to use these metrics to establish software quality. Even if these rules were usually oriented especially on the C++- source text, these find in the current literature their utility for Java or Smalltalk. As noted by Kan (Kan 2003, p.334), some of these metrics should be perceived more as a directive to design - OO and develop – OO, as metrics in the sense of a quantitative measurement. Image 4-2 shows an overview of these metrics together with the application rules.

As shown in (Lanza 2006, p. 18), the characterization, assessment and improvement of complex software systems is a cumbersome endeavor. Metrics can be applied to a wide variety of assessment measures. However, it should be considered how the data obtained can be made available to the observer. Lanza and Marinescu propose taking the inherent complexity of values of pure numbers through an appropriate view, to also allow to represent perceptibly complex contexts through adequate representation. They are limited to the visualization of programs and not of algorithms, the second form of software visualization. They describe the technique *polymetric views* to describe the software artifacts through metrics. Figure 8 shows a simple *polymetric view*.

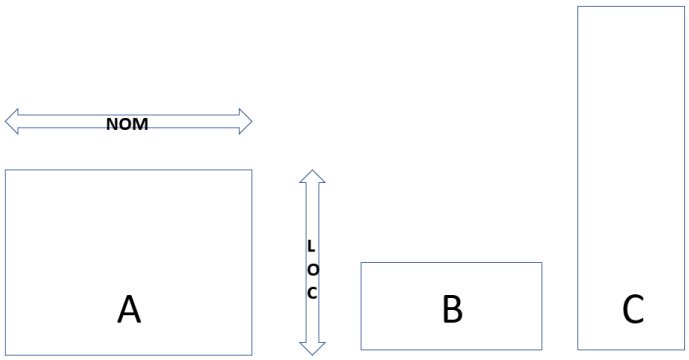


Figure 8 Simple Polymetric View representation (according to (LM06, p. 21])

It shows three classes (A, B, C) as rectangles whose metrics LOC and NOM (Number of Methods) are represented by the sides. Information conveyed by this representation is that B has the fewest lines of source code, while C has the most lines of source codes compared with A and B. In contrast, C has the lowest number of methods of all classes represented. From this, it can be deduced that C has the longest methods. Derived from this information, other measures may be implemented to improve software quality based on the fact that long methods have greater incidences of error than short methods.

Table 5 OO metrics and applications rules according Lanza&Marinescu

1.Sizes of average methods (LOC)	The nominal value must be less than 8 LOG for Smalltalk and 24 LOC for C ++
2.Average number of methods per class	The nominal value must be less than 20. Higher average values indicate that too much responsibility is concentrated in too few classes. The nominal value must be less than 6. More instance variables indicate that a class has more functionality than it should. The nominal value beginning with framework or root classes must be less than 6.
3.Average number of instance variables per class	The nominal value must be less than 6.
4.Depth of Inheritance Tree, DIT	The nominal value should be relatively high. This size is directly related to a high cohesion of classes in the same subsystem. If one or more classes of a subsystem do not interact intensively with other classes of the same subsystem, then it is better that these classes to be placed in another subsystem.
5.Number of subsystems / relations between subsystems	If method groups of a class use different sets of instance variables, then it is recommended to divide that class according to the usage of instance variables.
6.Class number / relations between the classes of each subsystem	The nominal value must be higher than t.
7.Usage of instance variables	The nominal value must be low (not specified in detail).
8.Average number of lines of comments (per method)	If a class is not used in different applications (especially abstract classes) it must be reconfigured.
9.Number of messages with problems per classes.	Should take a stable value throughout the entire development process. If not, then there is a suspicion that there is an incremental software development instead of an iterative software design and development.
10.Reuse frequency of classes	
11.Number of classes and methods that were rejected	

2.3.2.7 Integration in the software development process

After mapping out a broad software testing process, this cannot be independent from the software development process, the next subchapter deals with the dependencies and the connection points of the relevant methods of software development.

A. Model V (Sequential Development Model)

A traditional version to combine the sequential development of software with the testing process is represented by the V-model. The model shown in Image 9 claims that tests are conducted after software development. Unfortunately, this procedure leads to late recognition of errors in the early phase of development and therefore can be removed only with great effort. (Kurokawa & Shinagawa, 2008, p.37-38)

This problem is solved by the model shown in Image 10 that develops the original V-model. Thus, in every development phase of the original V-model, connections with the testing configuration are made, which not only improve the testing capabilities of the initial software configuration but can recognize configuration errors of tests already in the early stages of development. (Kurokawa & Shinagawa, 2008, p. 38-39)

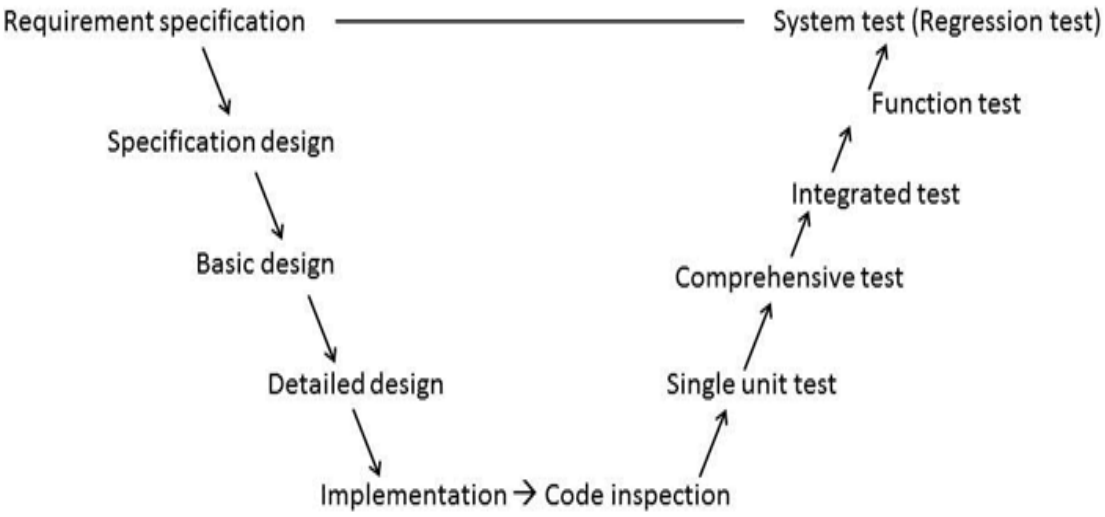


Figure 9 V-Model for software development and tests (Kurokawa&Shinagawa, 2008)

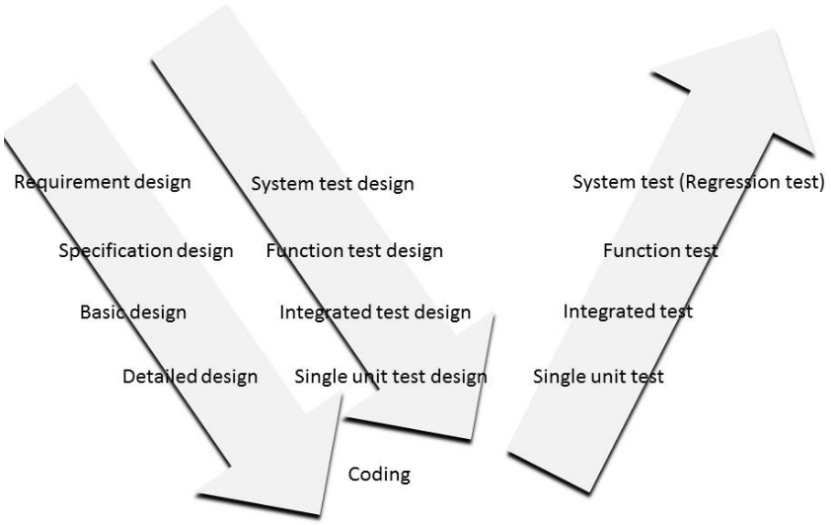


Figure 10 Model for software development and tests (Kurokawa&Shinagawa, 2008)

B. Iterative-incremental development models

While software development models undergo a complete design and specifications phase before proceeding with the implementation, iterative models like SCRUM, Rapid Application Developement (RAD) and Extreme Programming (XP) aim to complete software systems in short development cycles. For the software testing process, this means that systems that have been developed in an agile way (as shown in figure 11) currently undergo the process phases defined and levels of detail.

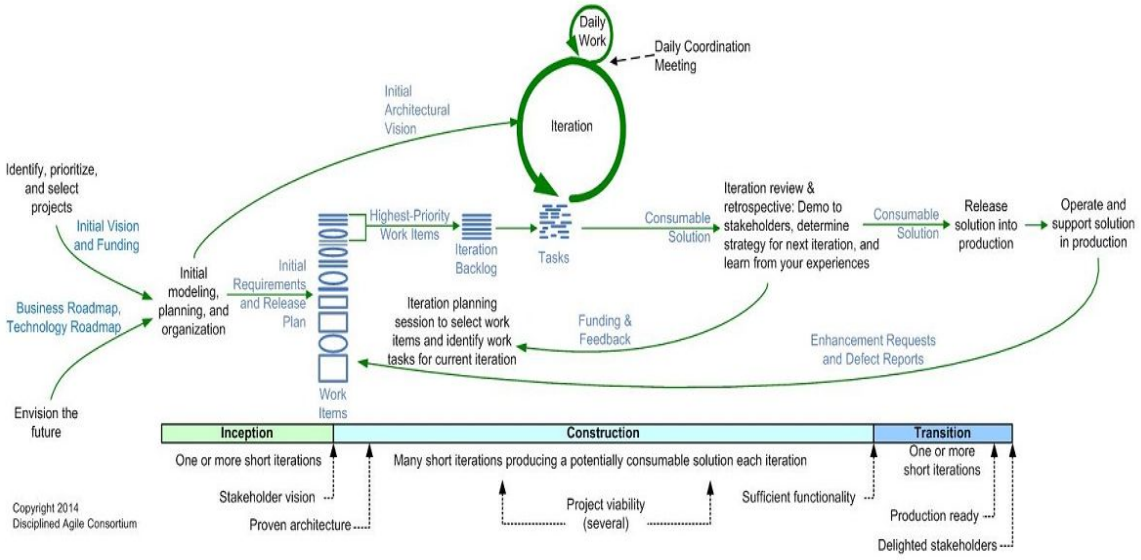


Figure 11 Agile test deployment (source picture http://www.ambysoft.com)

Because the software product is constantly growing through this procedure, an important role is played primarily by regression tests. This term means a certain amount of test cases that are repeated routinely to recognize unwanted changes to the software product.

At the end of this chapter the testing process, the structure and the phases of the testing process must be understood. The testing process was defined as specified by ISTQB and consists of the following six phases:

1. Test planning
2. Test analysis and configuration
3. Test performance
4. Assessment of final criteria and the report
5. Completion of testing activities
6. Test coordination

2.3.3. Process quality and product quality

The fact that product properties are created in the process of execution (technical processes of design and development) and are manifested in the process of use allows us to distinguish between the notion of **quality of the execution process** and **product quality**, between these existing an interdependence.

The quality of the process reflects the quality of the methods, techniques and tools through which the product is made, as well as the quality of the methods and means used to manage the process. It is determined by the ensemble of technological means used *to ensure* the quality of the product.

Product quality depends on the quality of the process through which it was achieved, and it represents the final expression of the execution process, thus synthesizing the technical, functional and economic level.

A level of quality achieved at a certain stage of the execution process influences the technical solutions in the next stage of the process, thus implicitly influencing the quality of the process. In addition, product quality– as the final expression of the execution process – reflects its quality.

Understanding these differences is very important from the point of view of software developers in setting product quality objectives and, separately, for the process. For a long time, developers considered only the technical aspect of software development, being concerned about the performance of the technical performance. This situation was determined by the fact that, on the one hand, *the quality requirements are not explicitly*

stated, and, on the other hand, quality achievement activities are perceived only through the control of the *process* of elaboration, not through the control of the product itself (Balog, 1994).

Product quality is evaluated by measuring internal attributes (static measures of intermediate products), by measuring external attributes (measuring product behavior) or by measuring attributes relating to quality in use. The overall objective pursued by evaluating is for the product to have the expected impact in the specified use context (Figure 12).

The quality of the process (the quality of any of the processes defined in SR ISO 12207:2000) contributes to the improvement of product quality, and product quality contributes to the improvement of the quality in use.

Consequently, evaluating and improving a process is a means of improving product quality, and evaluating and improving product quality are the means to improve quality in use. Similarly, evaluating quality in use provides the information needed to improve the product, and product evaluation provides the information needed to improve the process.

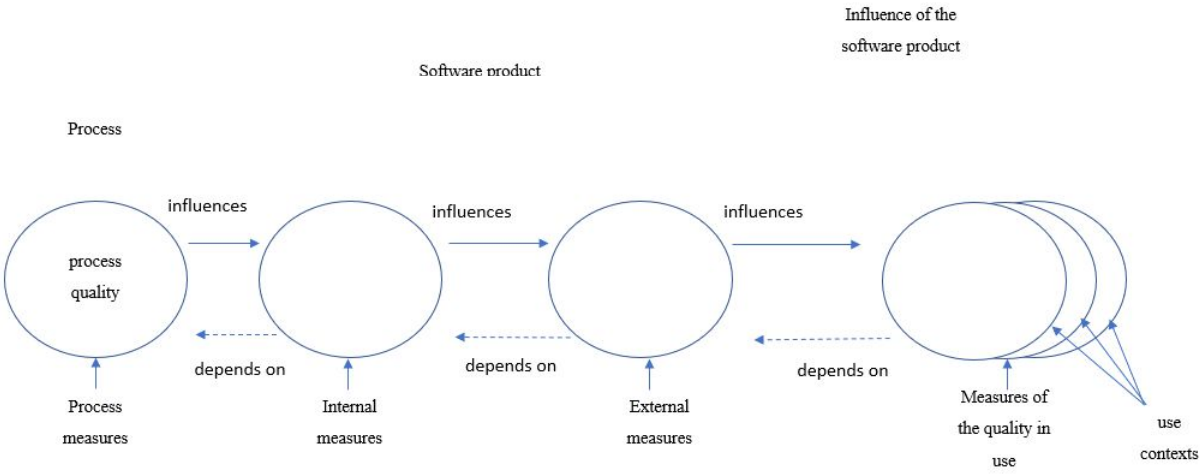


Figure 12 Product quality in the life cycle (source: adapted from ISO 9126)

2.3.4. Perspectives on Product Quality in the Life Cycle

Quality evaluation is a reflection of the various perspectives on quality. For the purpose of quality management at every stage of the life cycle, it is necessary to define these quality perspectives, and the changes that take place in product quality over the life cycle. The product quality perspective changes in the life cycle as follows:

- at the beginning of the life cycle, the quality specified by the “quality requirements” is addressed from an external perspective and from the point of view of the user;

- during the stages of the life cycle, quality (e.g., project quality, database quality, source code quality, etc.) is addressed from an internal perspective and from the point of view of the developer (designer, programmer, etc.);
- at the end of the life cycle, when the product is in service, quality is evaluated from an external and from the point of view of the user.

Figure 13 shows the quality perspectives at different phases of the life cycle of the product.

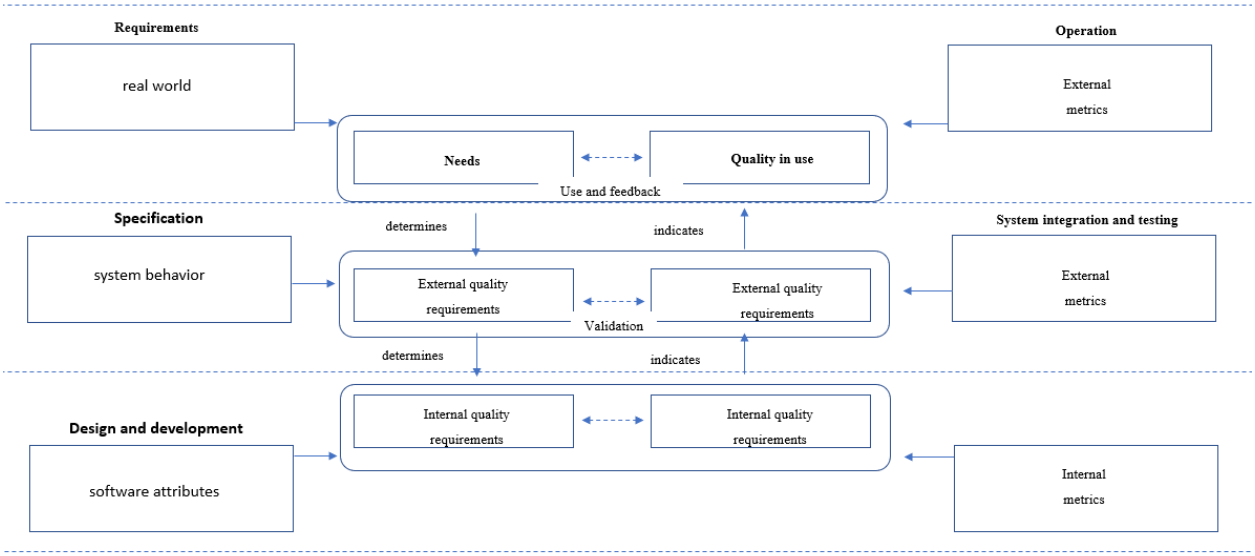


Figure 13 Perspectives on quality in the life cycle (source: adapted from ISO 9126)

Quality – seen as a *goal* to be achieved – is the necessary and sufficient quality that reflects the real needs of the interested parties.

The parties interested in a particular software product or system are of a wide variety: individuals (e.g., end users), organizations (e.g., company that develop the product / system, companies that purchase the product / system), society as a whole (e.g., state and regulatory authorities, the general public).

Different categories of interested parties have different needs that can be explicitly or implicitly stated. In general, implicit needs are determined by the context in which the product is used and are “user expectations” generated by the use of similar products and / or current procedures used. In many situations, the needs of different interested party categories may be in conflict.

However, the needs expressed by a user do not always reflect the real needs of the user, for at least the following reasons:

- the user does not always know his real needs;
- user needs may change after they have been declared;

- different users have different exploitation environments;
- it is almost impossible to consult all categories of users of a product / system in order to determine the full needs.

Thus, the quality requirements cannot be fully defined at the beginning of the product manufacturing process. It is necessary to understand as much as possible the real needs of the user and to represent needs in the form of *quality requirements*.

A software product is part of a system and therefore the quality requirements of the software product are related to the quality of the system of which the software product is part. System quality requirements include requirements for the quality of other system components (software, hardware, users).

Quality in use requirements define the quality level from the point of view of the end user. They have the following features:

- they are obtained (derived) from the needs of the context of use, such as: company policy, competitors, product scope, user features, etc.;
- they are used as criteria in user evaluation and validation activities;
- they are defined quantitatively in the appropriate documents (e.g., specification of requirements) by using the measures (metrics) of the quality in use;
- they are used to identify and define external quality requirements.

External quality requirements define the quality level from an external perspective that may be the one of the developer, the evaluator, or the buyer (the purchaser). They have the following features:

- they are obtained (derived) from a variety of sources, such as: quality in use requirements, the needs of the company developing the product, legal regulations, standards and recommendations specific to the type of product or scope, etc.;
- they are used as criteria in the verification and technical validation activities for the completed product;
- they are defined quantitatively in the appropriate documents (e.g., specification of requirements) by using external quality measures (metrics);
- they are used to identify and define internal quality requirements.

Internal quality requirements define the quality level from an internal perspective on the product (the developer's perspective). They have the following features:

- they are obtained (derived) from a variety of sources, such as: external quality requirements, company policy on product development, specific methods and techniques used by the developer, etc.;
- they are used for the purpose of specifying and planning the quality of intermediate results (products);

- they are used as criteria in the verification and evaluation activities of the quality attributes of intermediate results (products) in the product design and development process;
- they are defined quantitatively in the appropriate documents (e.g., specification of requirements) by using internal quality measures (metrics).

Quality as a goal does not necessarily mean perfect quality, but the quality required and sufficient for each use context specified when the product is delivered and used.

Internal quality represents the entirety of features of a product that determines its capacity (ability) to meet expressed and implied needs when used under specified conditions.

Internal quality is measured and evaluated against internal quality requirements, and is a reflection of the product design and development strategy, and predominantly takes into account quality sub-features and internal measures associated with the different attributes of the intermediate products (deliverables) obtained in the design and development process.

External quality (final product quality) is determined in each product design and development phase by estimating and predicting operations on quality features and sub-features using external measures for this purpose. Measurement and evaluation of external quality is performed by testing the product in a simulated environment and is based on the results obtained from internal quality measurement.

Quality in use is the user's perspective on product quality used in a specific environment and in a specific context. Quality in use is the combined effect of internal quality and external quality on the end user.

In order to determine quality in use, each product development stage involves estimates and predictions on the impact (effect) product use has. Quality in use measures the extent to which product users can achieve the proposed objectives in a specific environment (without using the product's properties).

Quality in use measurement and evaluation depends on the results obtained from external quality measurement, which in turn depends on internal quality measurement.

Internal measures are applicable to a non-executable product (specification of requirements, design specifications, source code, etc.) in the different phases of the design and development process.

External measures are applicable to an executable product in the final phases of the development process (component integration testing, system level testing), in product acceptance testing by the beneficiary, in the operation and maintenance processes.

Quality in use measures are applicable to the executable product used by different user categories for the purpose of achieving specific objectives in a specific environment and in a specific use context.

The definitions and significance of quality features and sub-features of software products are presented in Tables 6 and 7.

Table 6 Definitions of the quality features from ISO 9126-1

Features	Definition and meaning
FUNCTIONALITY	The capability of the product to provide functions that meet the expressed and implied needs when the product is used under specified conditions. This feature refers to what the product does to meet needs, while all other features relate to *how* and *when* it satisfies them.
RELIABILITY	Product capability to maintain a specified performance level when used under specified conditions.
USABILITY	The capability of the software to be understood, learned, used and considered attractive by the user when used under specified conditions. Users may be operators, end-users, and indirect users who are influenced or dependent on the use of the software. Usability refers to all types of usage environments that can influence the software, including preparation for use and evaluation of results.
EFFICIENCY	The capability of the product to deliver the appropriate performance in relation to the volume of consumed resources under specified conditions. Resources may include other software, hardware, materials (such as printing paper, floppy disks).
MAINTAINABILITY	Product capability to be modified. Changes may include product fixes, enhancements, or adaptations to changes in the environment, functional requirements and specifications.

PORTABILITY	The capability of the software to be transferred from one environment to another. The environment can include organizational environment, hardware, or software.

Table 7 The FUNCTIONALITY feature

Sub-features	Meaning
Adequacy	The capability of the product to provide an appropriate set of functions for specified operations and user objectives.
Accuracy	The capability of the product to provide accurate or acceptable results. Accuracy includes the required degree of precision of the calculated values.
Interoperability	The capability of the product to interact with one or more specified systems.
Security	The capability of the product to protect information and data so that unauthorized people or systems cannot read or modify them.
Compliance	The capability of the product to comply with the standards, conventions or legal regulations, and other similar prescriptions.

3. THE EXTENDED WEB ASSESMENT METHOD(EWAM)

3.1. Overview

EWAM (**E**xtended **W**eb **A**ssessment **M**ethod) is a method of evaluating and comparing the quality of e-commerce applications and websites. The method was developed between 1997 and 2001 by a group of researchers in the field of e-commerce at St. Gallen University in Switzerland, in collaboration with industry partners.

The main authors of the method are Prof. Petra Schubert and Prof. Dorian Selz who developed the conceptual framework of the method, and the results of the experiments performed were presented at international conferences or in articles in specialized journals (Schubert, 2003; Schubert and Dettling, 2001).

EWAM was developed based on the conceptual elements of a method previously developed at the Competence Center for Electronic Markets (CCEM) of the St. Gallen University (WAM – Web Assessment Method), to which other concepts and techniques have been added, such as:

- the client-oriented approach to developing and delivering products and services by electronic means;
- the phases of transactions on e-markets;
- electronic marketing;
- Technology Acceptance Model (TAM);
- the concepts of the theory of reasoned action (TRA – Theory of Reasoned Action).

The method responds to a major problem related to the evaluation of e-commerce applications: which are the most relevant criteria to ensure the success of an e-commerce system. The characteristic features of this method can be summarized as follows:

- it defines an evaluation grid that measures the quality and success of existing applications;
- it defines sectoral profiles that allow comparison of the target website with a medium-level profile or a high-profile profile (characterized by the best practices);
- it addresses the Internet-specific aspects;
- the evaluation is done from the client's perspective;
- it provides evaluators with an online tool for data collection and evaluation.

The EWAM method has been the subject of many experiments on e-commerce applications and services (B2C, B2B), and is commonly used in research, training and consultancy activities (Schubert and Dettling, 2001; 2002; Schubert and Selz, 1999; Schubert and Leimstoll, 2001; Selz and Schubert, 1997; 1998).

EWAM is a complex method that was created for a detailed analysis of a sample of applications and websites in a particular sector. In this respect, the method is not intended for the mass evaluation of websites.

3.2. Theoretical Foundations of the Method

3.2.1. The WAM Method

The WAM method at the origin of the EWAM method was developed in 1997 at the Competence Center for Electronic Markets (CCKM) of the St. Gallen University. The method defines an evaluation grid and a set of criteria for assessing the quality and success of e-commerce applications.

In addition to the client(user)-centered orientation, this method takes into consideration the success of the implementation of products and services specific to the

electronic environment. Schubert (2003) considers that the following paradigms are essential for the success of digital economy activities.

(a1) Electronic Markets and Transaction Phases

WAM takes into account the classic phases of electronic market transactions: information, negotiation and contracting, acquisition and payment. A fourth element called the "community component" is added as a link between the purchase transaction and the required trust in the virtual environment.

The content of the three classic transactions phases is as follows:

- *Information*. The client collects information about products and services of interest to him, searching for potential suppliers, prices and conditions.
- *Negotiation and contracting*. In this phase, the supplier and the client negotiate in order to reach an agreement or contract that sets out all the details of the transaction: product specifications, payment methods, delivery mode, etc.
- *Acquisition and payment*. At this stage the product / service is purchased by the client and the payment is made. During this phase the delivery (physical or virtual) of the ordered product takes place.

There may also be interactions such as, for example, warranty claims, service, assistance, and so on. Thus, the fourth phase of the transactions called "*After-sales services*" is added.

The "*Community component*" or the *customization* component, as it is called by Schubert and Gingsburg (2000), is a concept that serves as the essential link between two transactions and is formed in the process of communication between clients and between client and supplier.

Communities based on shared interests between members create a certain level of trust between them and facilitates the creation of a favorable environment for e-commerce. Collecting information from a specific community of interests facilitates the customization of the user interface and the product that is being offered.

Virtual communities that establish standards between their relationships inspire trust and allow for the creation of "*trusted intermediaries*". They guarantee generic services such as contracting, making payments, logistics and security of transactions, etc., and transforms the anonymity and (potential) disruption of the Internet into an electronic market with identified clients and transactions that can be recorded.

(a2) The Characteristics of the Electronic Environment.

For marketing issues, WAM emphasizes the special features that are inherent in the Internet environment. In this sense, some of the evaluation criteria are derived from the characteristics of the electronic environment, such as: hypermedia presentation, database interface, 24-hour access, anonymity, ubiquity, client integration and asynchronous communication.

(a3) Performance Marketing.

The basic idea of performance marketing is not to limit the activity to selling a product, but to provide the client with a complementary range of services to maximize client satisfaction. These additional services personalize the presentation of the product, and make it more attractive to the client, thus allowing for differentiation from the competition.

3.2.2. The Technology Acceptance Model

The EWAM method, fully revised in 2000, integrates a series of concepts and results of the TAM (Technology Acceptance Model) approach.

The TAM model was developed and published by F.D. Davis in his doctoral thesis (1985) at the Massachusetts Institute of Technology, USA. Using the TAM model, Davis described the effect of system features on the acceptance of new computer systems by users.

TAM is based on a psychological behavior pattern, later developed as a concept called “Theory of Reasoned Action” (TRA).

The TRA concept is based on the assumption that individuals act rationally, and take into account the implications of potential actions before engaging in a particular behavior. The individual will engage in a certain behavior if it leads to positive results. Behavior is influenced by the person’s attitude and subjective norm (Figure 14).

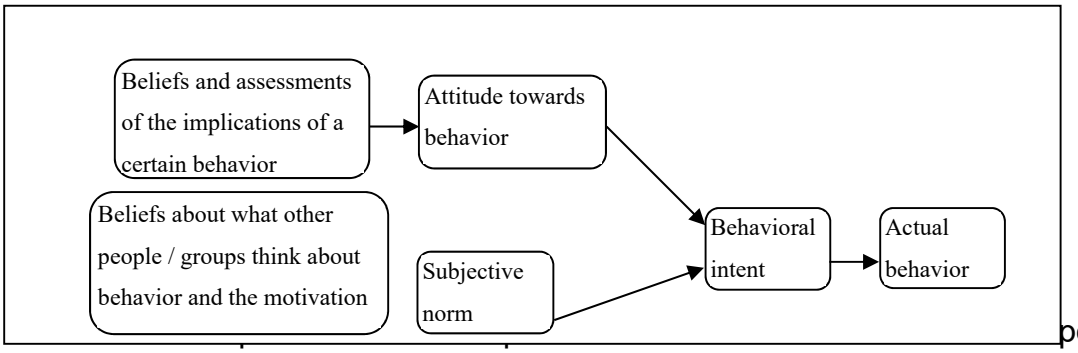


Figure 14 Motivated action theory

portant role in ame causal chain as defined in the TRA, namely: “Attitude” → “Intent” → “Behavior”.

On this basis, Davis suggested for the TAM model a causal diagram close to TRA (see Figure 15).

From Davis's experiments (by collecting data using questionnaires), he confirmed the following hypotheses:

- (H1) "Attitude" has a direct influence on "Actual system use";
- (H2) "Usefulness" has a significant effect on "Attitude towards use";
- (H3) + (H4) "Ease of use" has an effect on "Attitude" and "Usefulness";

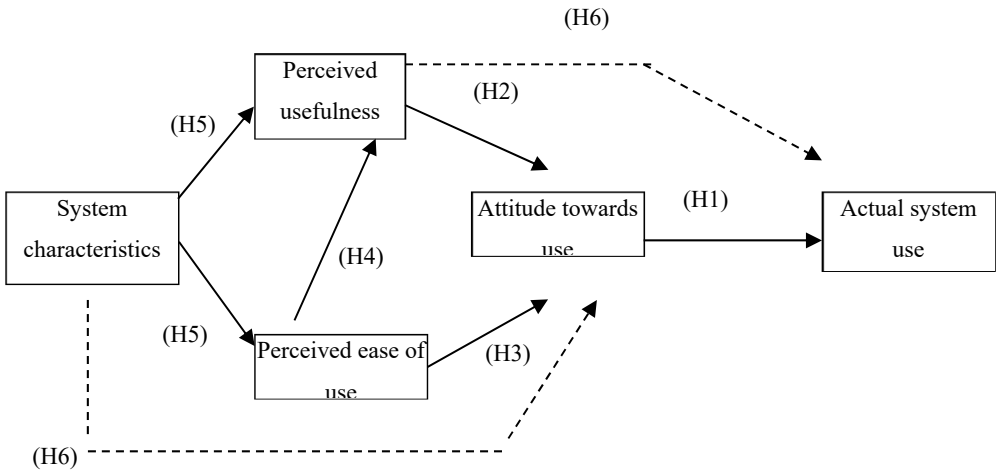


Figure 15 TAM causal diagram

⇢ Relevant causal relationship, but initially perceived as not relevant

→ Causal relationship initially perceived as relevant

The results of the experiments also provided new elements that were unexpected in this form, namely:

- (H5) "System features" have a significant effect on "Ease of use" but not on "Usefulness";
- (H6) "Usefulness" has a direct effect on "Actual system use";
- (H7) "System features" have a direct effect on "Attitude towards use".

The TAM model brings an important contribution to understanding the use, behavior and acceptance by users of new IT systems. The "Attitude towards use", "Perceived usefulness" and "Perceived ease of use" serve as a link between "System features" and the individual behavior associated with the use of a new system.

A critique to the TAM model refers to the fact that it has not also taken over the “Subjective norm” of the TRA concept that has an effect on the intent to use and accept information technologies. In the subsequent development of the EWAM method, the social influences have been reconsidered in the form of the “Trust” criterion.

3.3. Categories of Criteria and List of Evaluation Criteria

The EWAM method defines three *categories of criteria*: “Ease of use” (EOU*nn*), “Usefulness” (USEF*nn*) and “Trust” (TRUST*nn*).

Each category of criteria comprises a set of *criteria* that are allocated to one of the phases of the electronic market transactions (information, negotiation and contracting, acquisition and payment, after-sales services).

Also included are criteria for the after-sales service phase as well as for the “community component”. In addition, for the sake of an overall evaluation, the EWAM method includes a set of criteria that are applied to all phases of transactions (the so-called final evaluation). Figure 16 shows the general scheme of the EWAM method.

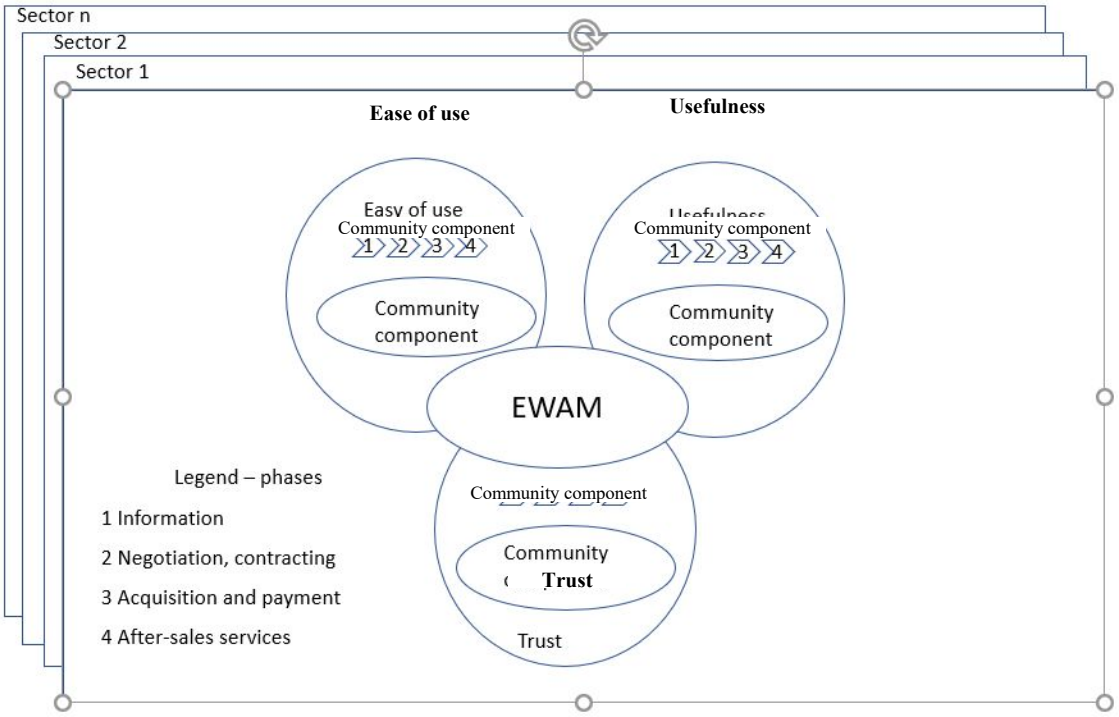


Figure 16 General EWAM diagram (source: adapted after Schubert, 2003)

The EWAM method uses the specific “profile” concept. This is a reference against which the results from evaluating an e-commerce application are compared. Three such profiles have been defined:

- *Sector profile*: means the scope of applicability to which the e-commerce application refers, such as providing financial services, supplying consumer goods, distributing digital goods (software), etc.
- *Organization profile*: means the attributes of the organization's website that provides e-commerce services in a specific sector;
- *"Best practice" profile*: means the set of methods, techniques, procedures, etc. demonstrated and proven to be the best in current practice and used in the provision of e-commerce services.

The evaluation criteria are formulated in general terms and are valid for any sector, but they are differentiated by the sector-specific importance. In this way, the conditions for identifying the reference sector and benchmarking of different e-commerce applications are created both within a sector and between different sectors.

The establishment of (referential) "profiles" allows the comparison of the quality of an e-commerce application with the average quality of a given sector, and, at the same time, the comparison with the best practices used in the sector.

Note that referentials can only be established after evaluating a sufficient number of e-commerce applications in a particular sector, and comparing the results obtained with client satisfaction criteria.

Tables 8 to 12 present the evaluation criteria used in the EWAM method for each of the phases of electronic transactions.

Table 8 EWAM criteria for the "Information" phase

No.	Criterion	Description
1	*User interface*	The criterion evaluates ease of use for both frequent users and new users. Includes page load time and guidance during the interaction process.
2	*Content structure*	The criterion measures ease of access, as well as impressions on the logical structure of content. Content tables, navigation frames, and site maps are typical examples of features that make navigating easier.
3	*Reasonable amount of information*	The amount of information refers to the margin of information about a company, products and services.
4	*Advantages of recording*	Many websites require clients to register or at least provide a minimum of personal information. On the other hand, they may have some advantages: (a) grant of credits; (b) discounts on the purchase of products;

5	*Possibilities to combine products and services*	The criterion refers to the possibilities of online bundling of offers (combining the company's products or products with goods and services offered by a third party). E.g.: combination of ticket sales and hotel booking.
6	*System availability / performance*	The criterion measures global availability (availability versus geographic aspects) and loading speed no matter where the client is.
7	*Savings to the client*	Electronic sales often reduce the cost of transactions so that e-commerce clients can benefit from lower prices.

Table 9 EWAM criteria for the "Negotiation and contracting" phase

No.	Criterion	Description
1	*The user (client) profile can be updated*	Transactions require the client to provide personal information (e.g., information on payment options). Storing this information allows its reuse in subsequent transactions.
2	*Placing an order is guided to the corresponding profile (customized services)*	To benefit from a high degree of customization of services, clients can provide additional information. On the other hand, the system can track user activity. A detailed user profile (age, gender, preferences, hobbies) helps to differentiate the client and allows for suggestions or special discounts.
3	*Product customization possibility*	Some clients may be interested in buying combinations of products (product systems) or just some product fragments (e.g.: parts of magazines and newspapers). The website must support the customization of products in accordance with the user's attributes.
4	*Transparent, interactive business rules integration*	The basic rules refer to the general delivery terms and conditions, warranties, returns of the product, etc. In this context, it is useful to have options or facilities (e.g., buttons) on the website to accept terms and conditions, and to guide the interaction.

5	*Implementation of security aspects*	Websites must provide reliable security features (such as digital certificates) or implement accepted standards in that field.
6	*Contact possibilities (support desk)*	The criterion refers to the different ways to establish communication with the seller. It may include the implementation of a helpdesk or call center. The website can offer (a) opportunities to write and read questions for general interest issues (FAQ) (b) feedback possibilities (e-mail or web formats).

Table 10 EWAM criteria for the "Acquisition and payment" phase

No.	Criterion	Description
1	*Easy select of generic services*	Generic services are software modules available across the web platform, and that have a unified interface. Generic services support an electronic transaction. Example: electronic payment services.
2	*Integration of generic services*	Good integration means that they are used whenever needed, providing the user with a consistent interface and routine operations.
3	*Effective use of the user profile*	During the phase, personal information about the client is requested. This information must be stored safely and must be available for later use.
4	*Tracking transactions*	Examples: direct access to your personal order information, order status, and so on.
5	*IT integration*	The criterion refers to the connection with the client infrastructure. Especially for SMEs, a link to the financial-accounting system can be of great help.
6	*Convenient after-sales support*	The website should support after-sales services (e.g., feedback formats, warranty formats).

Table 11 EWAM criteria for the "After-sales services" phase

No.	Criterion	Description
1	*Support services convenient for the client*	The website must offer facilities for the provision and monitoring of after-sale services.
2	*Client satisfaction with after-sales services*	The website should include facilities to determine client satisfaction (e.g., include online questionnaires that can be filled in by clients), including the publication of results.

Table 12 EWAM criteria for the "Community component" phase

No.	Criterion	Description
1	*Access of one or more communities*	The criterion refers to "virtual communities" (the union and communication between individuals who share common values and interests and use the electronic environment to get in touch with each other; communication is time- and space-independent). These communities are more or less attached (true) to a website. A high value of the criterion indicates that there is a good link between the product offer and the website community.
2	*Uniqueness / originality of information*	The criterion refers to the value of information that can be obtained from the community area, given that information is difficult to obtain from other sources. A community that includes experts who actively contribute to the community area can provide information that cannot be obtained from other sources.
3	*Appropriate number of members*	The value of a community consists of its members. There must be a few "core members" who specifically devote their activities to the community. The higher the number of members, the more likely it is to have good questions, answers, reviews and other such contributions.
4	*Well-implemented customization and*	When joining a community, people are usually looking for partners with similar tastes and interests.

	collaborative filtering mechanisms	There are two kinds of profiles that can be stored in the database: personal information about interests and tastes (client input), and user interactions (login activities). The second category contains dynamic information from which patterns of user interest and behavior can be derived.
5	*The member can choose how to appear in the community*	Some websites allow a person to choose his or her representation, called avatars, which appear in the form of animals, people, comic characters.
6	*The private character is sufficiently protected*	Sometimes you can have access to a community without providing personal information. There are clients who prefer anonymity in the virtual space.
7	*Real added value perceived by the community member*	The criterion evaluates the value of community membership. In addition to the information that can be gained from the community area can also appear as added value (e.g., establishing personal relationships with other members). Specific experiences about shared life are the basis of community relations.
8	*Customized "push" mechanisms to establish a relationship with the client*	The "push" technique is characterized by the automatic transmission of information to the client (or a member of the community). It can choose between receiving up-to-date information and receiving information without requesting it. The mechanism must be customizable by the client.
9	*"Pull" mechanisms to establish a relationship with the client*	The client / community member actively seeks information when needed. As a rule, the effects of "pull" actions make up the results of the facilities offered by the supplier of products / services (list adding, price reductions, etc.).

3.4. EWAM Procedures and Algorithms

The EWAM method is based on a double evaluation of each criterion. In a first step, the evaluator assesses the importance of each criterion. The next step is the actual evaluation of websites in the analyzed sector.

When determining the levels of importance of the criteria and when evaluating, a four-value scale (++, +, -, - -) is used. The significance is presented below. **N.A.** (not applicable) is used in situations where a criterion is not relevant or does not apply in a particular context.

Symbol	Meaning
++	very important
+	important
-	less important
- -	unimportant
N.A.	not applicable

These symbols are used in the data collection process with the help of questionnaires filled in and received from clients. In the calculation and determination of the EWAM method indicators it is necessary to convert the symbols into numbers.

Aggregation of importance ratings valued by evaluators in stage one is a particularly important requirement for generating results, given their calculation algorithms; a low importance ("- -") largely annihilates evaluations in stage two, and diminishes the impact of the assessment of the criterion in the overall rating.

The basic algorithms of the EWAM method provide for the multiplication of the importance rating with the average of the evaluations for each criterion, and their aggregation across the 6 categories (phases) corresponding to the transaction phases, *After-sales services*, *Community component*, and *Final section*. The calculation procedure has the following advantages:

- it eliminates the extreme values of client expectations (importance) for a certain criterion;
- it takes into account the different levels of experience of evaluators.

The following describes how to calculate the resulting rating of the evaluation – based on the EWAM indicators included in Table 13

The indicators allow the comparison of any website examined with the media and the best industry practice, including the rating of the importance of criteria for the "Sector profile". Similarly, the other two profiles ("Organization profile" and "Best practice profile") are calculated.

Table 13 EWAM indicators

Criteria: X_i	Importance: W_i value range (-2, +2)	Importance: Wg_i value range (0, 1)	Evaluation: $\overline{r}_i$ value range (-2, +2)	Weighted value: R_i $R_i = Wg_i * \overline{r}_i$
X_1	W_1	Wg_1	$\overline{r}_1$	R_1
…	…	…	…	…
X_{26}	W_{26}	Wg_{26}	$\overline{r}_{26}$	R_{26}

The criteria (X_i, $_{i=1..26}$) are grouped into six categories (K_k, $_{k=1...6}$) corresponding to the three phases of the transaction (*Information; Negotiation and contracting; Acquisition and Payment)*, the *"After-sales service"* phase, the *"Community component"* and the *"Final section"*, as presented on the data collection questionnaire (namely 8 criteria for K_1, 2 criteria for K_2, 3 criteria for K_3, 2 criteria for K_4, 4 criteria for K_5, 7 criteria for K_6).

(a) Transforming the importance of a criterion

For evaluating the individual criteria and their importance, a scale of integers ranging from (-2) to (+2) is used. For the convenience of calculating and avoiding the multiplication of two negative numbers, the importance of each W_i criterion is transformed (normalized) in the range $(0,1) \subset Q$ using the following formula:

$$Wg_i = 1/4 (W_i + 2)$$

(b) Averaged evaluation of each criterion (result)

$$\overline{r}_i = 1/m \sum_{j=1}^{m} r_{ij}$$

where:

m is the number of evaluators evaluating criterion X_i;

r_{ij} is the individual result of the criterion in the $(-2, +2) \subset Z$ interval.

(c) Multiplication of the result by the rating of the importance of each criterion

The final result for each criterion is obtained by multiplying the result (average of its evaluations) with the rating (level of appreciation) of importance.

Thus, a criterion will have the maximum rating only when the level of appreciation of the importance is maximum (in other words when $Wg_i = 1$). With lower ratings of importance ($1 \geq Wg_i \geq 0$), the final result of a criterion decreases.

$$R_i = \bar{r}_i * Wg_i$$

(d) Elementary evaluations aggregate per each category

The result for each category (K_k, $k = i...6$) is obtained by adding the weighted value R_i per category. For example, for the *Information* phase (K_i) and the *Acquisition and payment* phase (K_2), the calculation method is as follows:

$$K_1 = \sum_{i=1}^{8} R_i$$

$$K_2 = \sum_{i=9}^{10} R_i$$

(e) Calculation of the percentage of reaching the maximum rating for K_i i = I…6

(e1) Calculation of the minimum R_{MINi} and maximum R_{MAXi} evaluations for a criterion with the importance given in the interval (0,1)

$$R_{MINi} = Wg_i * -2 \quad -2 \leq R_{MINi} \leq 0$$

$$R_{MAXi} = Wg_i * 2 \quad 0 \leq R_{MAXi} \leq 2$$

(e2) Calculation of the minimum K_{MINi} and maximum K_{MAXi} for a category, with examples for the Information *phase (K_1):*

$$K_{MIN1} = \sum_{i=1}^{8} R_{MINi}$$

$$K_{MAX1} = \sum_{i=1}^{8} R_{MAXi}$$

(e3) Calculation of the percentage of reaching the maximum rating

$$K\%_k = \frac{K_k + K_{MAXi}}{K_{MAXi} + (K_{MINi} * -1)} * 100\%$$

or simplified:

$$K\%_k = 0.5(\frac{K_k}{K_{MAXi}} + 1) * 100\%$$

(f) Transforming $K\%_k$ in the interval (-2, + 2) – the final result of each category:

$$KR_k = (\frac{K\%_k}{100} * 4) - 2 \quad k=1..6$$

A first analysis is performed on the final result of each KR_k k = 1…6 category. A value in the interval (-2, + 2) – for example, 0.96 for the *Information* phase – indicates that in the sector average the phase was evaluated as relatively good (on the rating scale, a value KR_k = 1 corresponds to the "good" rating).

(g) Comparing the results of the individual categories with the rating of importance

Another analysis consists in comparing the final result of a category (KR_k) with the average rating of its importance (KW_k), which indicates the degree of response of the sector to the client's expectations for the k category (phase). For example, for the *Information* phase or *Final section*, the average value of the importance rating is:

$$KW_1 = 1/8 \sum_{i=1}^{8} W_i$$

$$KW_6 = 1/7 \sum_{i=1}^{7} W_i$$

(h) General rating

It is the final result of a profile (PR). It is calculated as the sum of the six categories (KS) in relation to the theoretical maximum result of the respective profile.

h1) Calculation of the sum of all categories

$$KS_o = \sum_{k=1}^{6} K_k$$

where $o, (o = i...3)$ indexes the three profiles (Sector profile: $o = 1$)

h2) Calculation of the percentage of reaching the maximum rating for KS

The calculation is similar to the one in point (e3), but for all six categories.

$$KS\%_o = \frac{KS_o + RS_{MAXo}}{(RS_{MAXo} + RS_{MINo} * -1)} * 100\%$$

where:

RS_{MAXo} is the theoretical maximum of the rating for all criteria in profile $_o$

RS_{MINo} is the theoretical minimum of the rating for all criteria in profile $_o$

h3) Transforming KS% 0 in the (-2, +2) interval

$$PR_o = (\frac{KS\%_o}{100} * 4) - 2$$

For the sector profile, a value $PR_0 = 1.05$ has the meaning of "good" on the scale (-2, + 2), that is, the presence of websites in the sector average was appreciated as good.

4. EXPERIMENTING WITH THE EWAM METHOD

4.1. Objectives pursued by experimentation

The objectives of experimenting with the EWAM method are (Balog et al., 2003):

- verifying the applicability of the method for measuring and evaluating websites in the field of e-commerce in Romania and Austria – fashion sector;
- verifying and validating website evaluation criteria specified in the EWAM method;
- formulating new requirements on the evaluation criteria of websites, in general, and of e-commerce websites for fashion sector, in particular.

4.2. Conditions of the experimentation

The aim of the following quantitative study is to investigate if the EWAM method is applicable for measuring and evaluating websites in the field of fashion e-commerce in Romania and Austria, if the method can be nowadays extended due modern technology and internationalization/globalization.

The empirical section based on The Extended Web Assessment Method examines and evaluate the online shops of Orsay, Zara, H&M and Mango websites based on

interviewing people from Austria and Romania. This is a very ambitious and labor-intense work because the assessors should meet certain criteria:

- They need to understand the criteria of the Web assessment form very well, hence they must undergo a thorough instruction
- They must be experienced Web users
- They must take the time to go through all four transaction phases for each Web sites assessed (including delivery and payment!)

The survey:

The survey was conducted anonymously via www.onlineumfrage.com (https://www.umfrageonline.com/?url=survey_det&uid=1045452), and its aim was to answer the following question:

Which new requirements on the evaluation criteria of websites, in general, and of e-commerce websites for fashion sector improve the quality of e-commerce web site from user perspective?

The study is divided into two distinct parts, and the first one consists of 31 questions – specific EWAM method. Their target is to verify and validate the EWAM the methods criteria's for measuring and evaluating websites in fashion e-commerce in Romania and Austria.

EWAM is a complex method that was created for a detailed analysis of a sample of applications and websites in a particular sector in all electronic transactions phases: Information phase, Negotiation and contracting phase, Acquisition and payment phase, After-sales services phase, Community component phase. The EWAM method uses the specific "profile" concept. This is a reference against which the results from evaluating an e-commerce application are compared. One of such profiles have been defined "Best practice" profile – which means a set of methods, techniques, procedures, etc. demonstrated and proven to be the best in current practice and used in the provision of e-commerce services.

The establishment of (referential) "profiles" allows the comparison of the quality of an e-commerce application with the average quality of a given sector, and, at the same time, the comparison with the best practices used in the sector.

In addition to the client (user)-centered orientation, this method takes into consideration the success of the implementation of products and services specific to the electronic environment.

The second part consists of 10 questions that will be performed to find if the EWAM methods criteria can be nowadays improved to evaluate e-commerce systems, especially in fashion sector, if the users' needs are changed due the modern technology, internationalization/globalization, if the user needs are different – compare results from two Europe countries –Austria (central Europe) and Romania (eastern Europe).

The answers from the questions: "Does the site offer the option to view non-English pages? Is text correctly displayed and that sentences are grammatically correct? Does the site respect cultural differences between the home and foreign country? Calculates the system correctly taxes and exchange rates for the regions concerned?" will conclude if the e-commerce systems operate in selected target countries as well as it does in its home country, across geographic boundaries or technically performance. It is a good point with a potentially criteria to the method.

The answer to the questions: "Exists the possibility to select the products according to type of outfit? (e.g.: casual, elegant, clubbing, sport etc.) Exist a "complete the outfit" tool with complementary products to the current one? (Preferably chosen manually by a fashion editor) Has a commercial description done by fashion editor/specialist? (Emphasized as such – "what the specialist says", "fashion editor's recommendation" etc.) Is the presence of any of the following facilities: 360 degrees interactive pictures with the product, virtual fitting room or catwalk? Has the evaluated web-site an app? Have you any other recommendation to improve the website evaluation? Please enter your recommendation" will evidence which are the particularly needs to for fashion e-commerce web sites, if the user needs are changed due modern technology and depend of this can be conclude if the 'best practices' profiles can be extended with new criteria on evaluation of fashion e-commerce websites due modern technology and internationalization/globalization.

The questionnaire in the form of an Excel file was filled in by each evaluator. The questionnaires were centralized and processed following the procedures and algorithms for calculating the EWAM indicators.

4.3. Experimentation Methodology

The evaluation of the selected e-commerce websites was carried out analogously to the evaluation methodology presented by the authors of the EWAM method (Schubert, 2003).

Before the evaluation was started, the evaluators were selected, prior training was carried out for the purposes of the evaluation and how the questionnaire was filled in.

Table 14 provides a questionnaire for evaluators to assess the importance of criteria in the analyzed sector (column *Imp*), and to record the results obtained from website evaluation (columns *S1*, *S2*, *S3* and *S4*)[4].

For ease of further processing, the corresponding numbers (-2 for "--", -1 for "-", 1 for "+", 2 for "++") were used instead of the symbols used in the online EWAM questionnaire.

The presented questionnaire is suitable for rapid calculations of indicators and levels of importance according to EWAM algorithms.

Table 14 The format of the EWAM questionnaire used in experimentation

Phase / Criterion	ID	Imp	*S1*	*S2*	*S3*	*S4*
1. Information phase						
Website accessibility and product offerings	EOU01					
Content structure	EOU02					
Amount of information	EOU03					
Quality of content	USEF01					
Transfer by cost of the supplier's cost-related benefits	USEF02					
Packages of products and services	USEF03					
Recommendation systems	USEF04					
Hypermedia usage	USEF05					
2. Negotiation and contracting phase						
Product and service order procedure design	EOU04					
Models and pricing methods	USEF06					
3. Acquisition and payment phase						
Integration of generic services	EOU05					

[4] for the purpose of confidentiality, symbolic names for each site (S1, S2,…) are used throughout the paper. There is no association between the previously defined site order and the serial number of the S_i symbolic name.

Integration of e-commerce applications into the client system	USEF07					
Order traceability	USEF08					
4. After-sales service phase						
Access to client support	EOU06					
Client support utility	USEF09					
5. Community component						
Access to a virtual community	EOU07					
Usefulness of relationships in the virtual community	USEF10					
Usefulness of content accessible through the virtual community	USEF11					
Power of client influence in the virtual community	USEF12					
6. Final section						
System availability	EOU08					
User interface design	EOU09					
Productivity increase by reducing the time consumed	USEF13					
Interaction	USEF14					
Customization features	USEF15					
Business partner (supplier) trust	TRUST1					
Website respectability and legal situation	TRUST2					

For a better interpretation by the evaluators of the significance of the values provided for each criterion included in the questionnaire, a help with the possibility of retrieval according to the criterion code was made and provided to them.

In **Appendix 4.3** to this chapter, the significance of the evaluation values and the conditions for a website to receive the maximum (ideal) rating for each criterion in the questionnaire are presented.

4.4. Results Obtained; Analysis and Interpretation of Results

The evaluation results for the four websites selected are summarized in Table 4.8. All values entered in the table are within the range (-2, + 2).

The "*Importance*" column contains the result of the data processing relative to client expectations (column *Imp* in the questionnaire). The *Sector profile* column shows the result of applying the EWAM algorithms previously presented to determine the sector profile and allows the site's position to be evaluated in the analyzed sector.

		S1	S2	S3	S4	
	Inportance	www.zara.com	www.mango.com	www.hm.com	www.orsay.com	Sector profile
Information phase	0,60	0,69	0,64	0,44	0,83	0,65
Negotiation and contracting phase	0,72	0,88	0,92	0,54	0,73	0,77
Acquisition and payment phase	0,63	0,68	0,46	0,63	0,73	0,62
After-sales service phase	0,49	0,82	0,50	0,14	0,82	0,57
Community component	0,43	0,57	0,50	0,33	0,36	0,44
Final section	0,68	0,80	0,76	0,52	0,84	0,73

Figure 17 Evaluation results

Regarding the results from the evaluation, the following comments are made:

- from the point of view of client expectations, the *Final section* was found to be the most important (0,80) – meaning that most relevant criteria are important (they meet the conditions for at least the "+" rating). In fact, the section contains the overall criteria that apply to all phases (system availability, user interface, interactivity, etc.), and which are very important to the client of the e-commerce website.
- in the order of importance, the *Information* phase (value of 0.83) and the *After-sales service* phase (value of 0.82) follow.
- the *Community component* category obtained a very low rating value (value of 0.33). An explanation is probably the still weak state of their establishment, and insufficient user knowledge of the potential of communities as a support for obtaining recommendations, the combination of products – services, price influence, etc. Evaluations for this category of criteria are "weak" for all websites analyzed.
- in relation to the level of importance given to the *Final section*, the four websites analyzed had ratings ranging from 0.52 to 0.84, which is a "good" rating, with close results to the 0.73 sector profile;
- in relation to the level of importance given to the *Information* phase, sites achieved ratings 0.44, the "satisfactory" rating (the percentage of reaching the maximum rating ranges between 59,3%);
- for the *After-sales service* phase, website S3 achieved ratings ranging from 0.14, which is a "weak" rating – with close results to the sector profile (0.57);

- the close ratings obtained by all four websites for the *Acquisition and payment* phase stand out, but also the differences between *S1* and *S2* compared to *S3* and *S4* (qualified as “neutral”) in the *Negotiation and contracting* phase, with implications for the sector profile rating to this category of criteria.

In Figures 18 to 19, the results of the evaluation on categories (phases) for the sector profile and the S2 organization profile relative to their importance (client expectations) are compared.

Depending on the position of the result in the graph’s quadrants, certain strategies to be followed to remedy unfavorable situations must be followed. The presence of the *After-sales service* and *Community component* in the lower left quadrant for the *S3* organization attests to the “poor” rating and differences compared to client expectations.

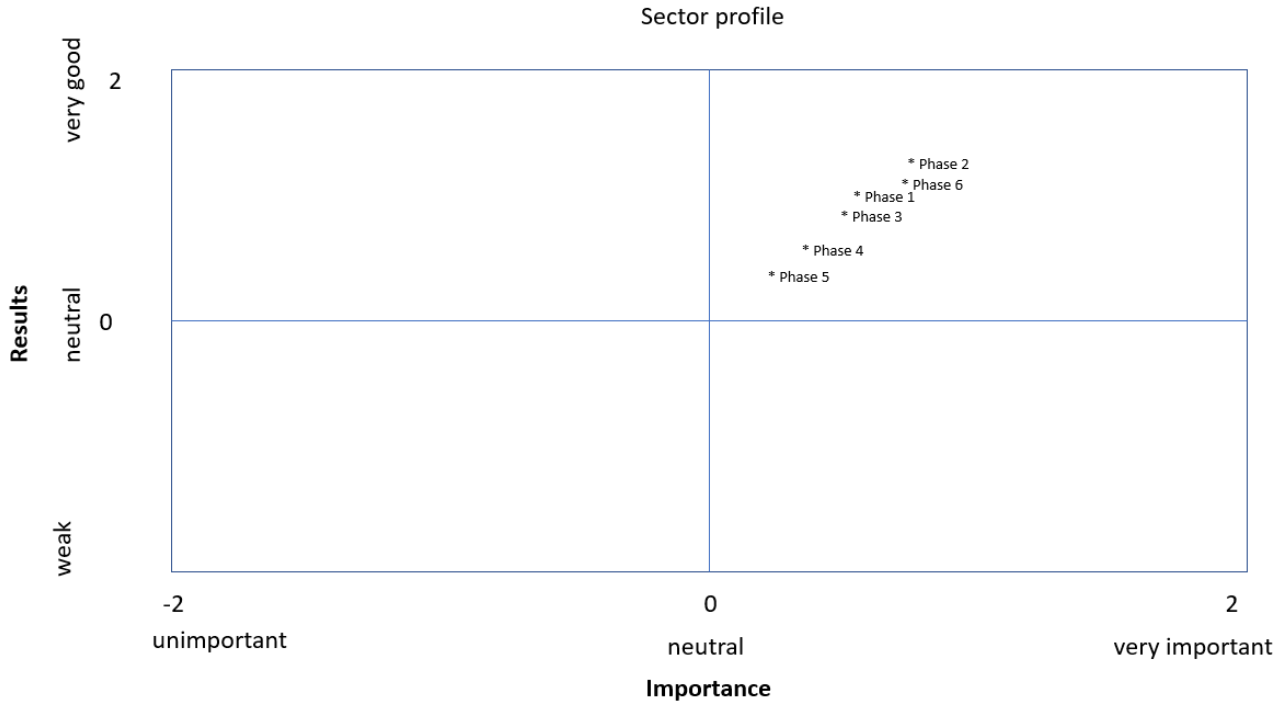


Figure 18 Result ratings by categories (phases) for the sector profile

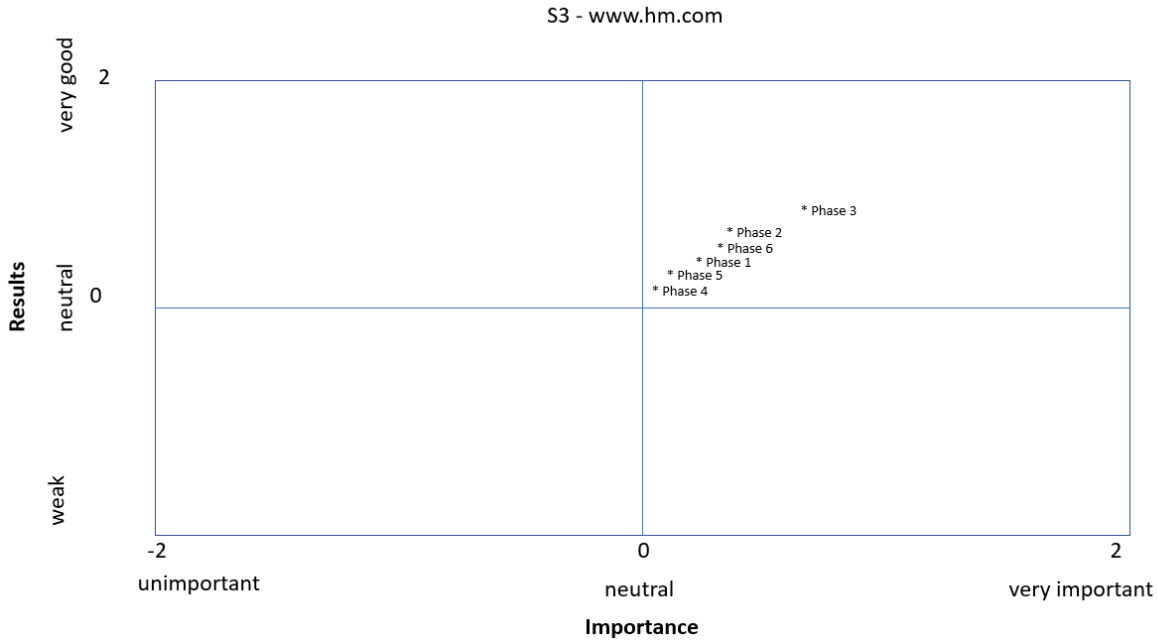


Figure 19 Results ratings by categories (phases) for the S3 organization

Table 15 lists the recommended EWAM generic strategies for different cases.

Table 15 Generic EWAM strategies

Strategy	Results
Strategy redesign	The presence in the upper left quadrant indicates good or very good results for a rather unimportant category (phase) (negative values of importance). Probably the available resources are not effectively used.
Maintaining the strategy	The presence in the upper right quadrant indicates good or very good results for an important or very important category (phase).
Not immediate improvements are necessary	The presence in the lower left quadrant indicates poor or very poor results for a rather unimportant category (phase) (negative values of importance).
Improvements are needed	The presence in the lower right quadrant indicates poor or very poor results for an important or very important category (phase).

Table 16 presents the comparative results of the *S1* organization rating (which, besides the general rating, obtained the highest rating for the *After-sale service* phase and the *final section* phase) with the sector profile and the “Best practices” profile. For the

"Best practices" profile, the data provided by the authors of the EWAM method (Schubert, 2003) were used.

The *S1* organization has comparable results to the sector profile for most of the categories (phases), with the largest difference for the *Community component* – which also achieved the best rating (0.57). This has results close to the "Best practices" profile, with the exception of the *After-sales service* phase, where the difference reveals the "good" rating for the phase-specific criteria.

Table 16 Comparative results for the evaluation of S1

S1 results Phase	Importance	Organization profile			"Best practices" profile	Sector profile
			Difference		(CBP)	(PSC)
			CBP	PSC		
1. *Information* phase	1.01	0.69	0.02	0.04	0.67	0.65
2. *Negotiation and contracting* phase	1.02	0.88	0.27	0.11	0.61	0.77
3. *Acquisition and payment* phase	1.76	0.68	0.23	0.07	0.45	0.62
4. *After-sales service* phase	1.11	0.82	-0.01	0.25	0.83	0.57
5. *Community component*	-0.27	0.57	-0.11	0.13	0.68	0.44
6. *Final section*	1.50	0.80	-0.24	0.07	1.04	0.73

Figure 20 and 21 presents graphically the results of the website evaluation for *S2* (Organization profile), and its comparison with importance.

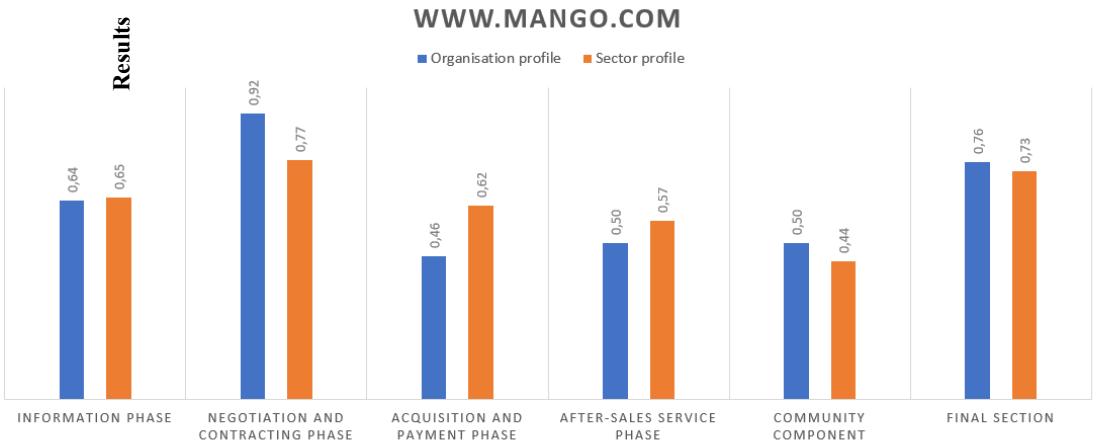


Figure 20 Graphical representation of S2 evaluation results

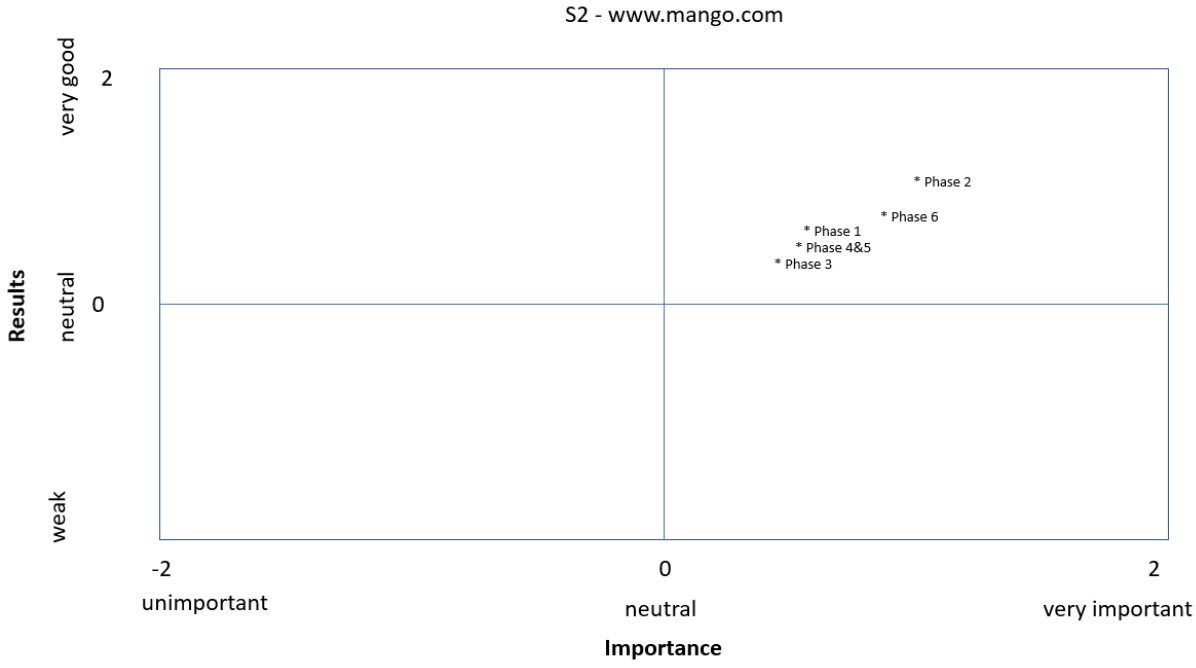


Figure 21 Graphical representation of S2 evaluation results

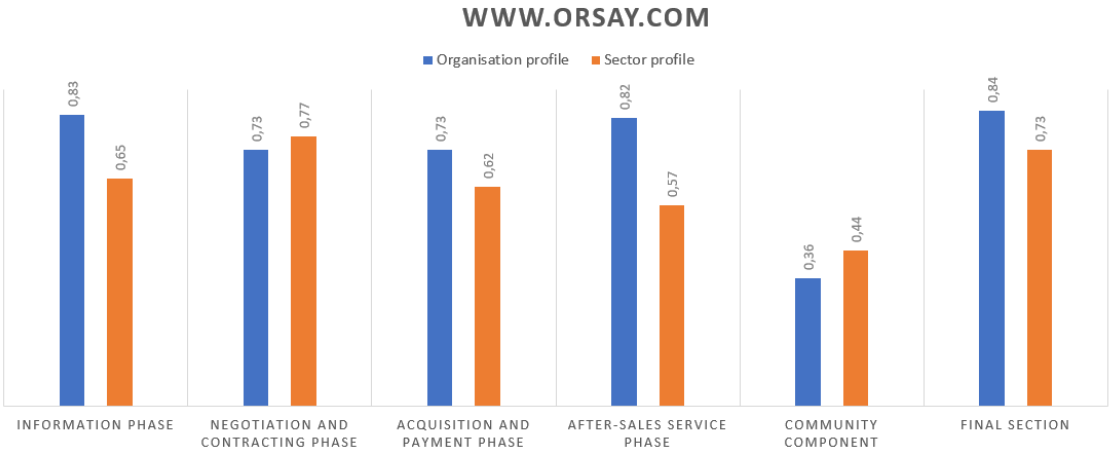


Figure 22 Graphical representation of S4 evaluation results

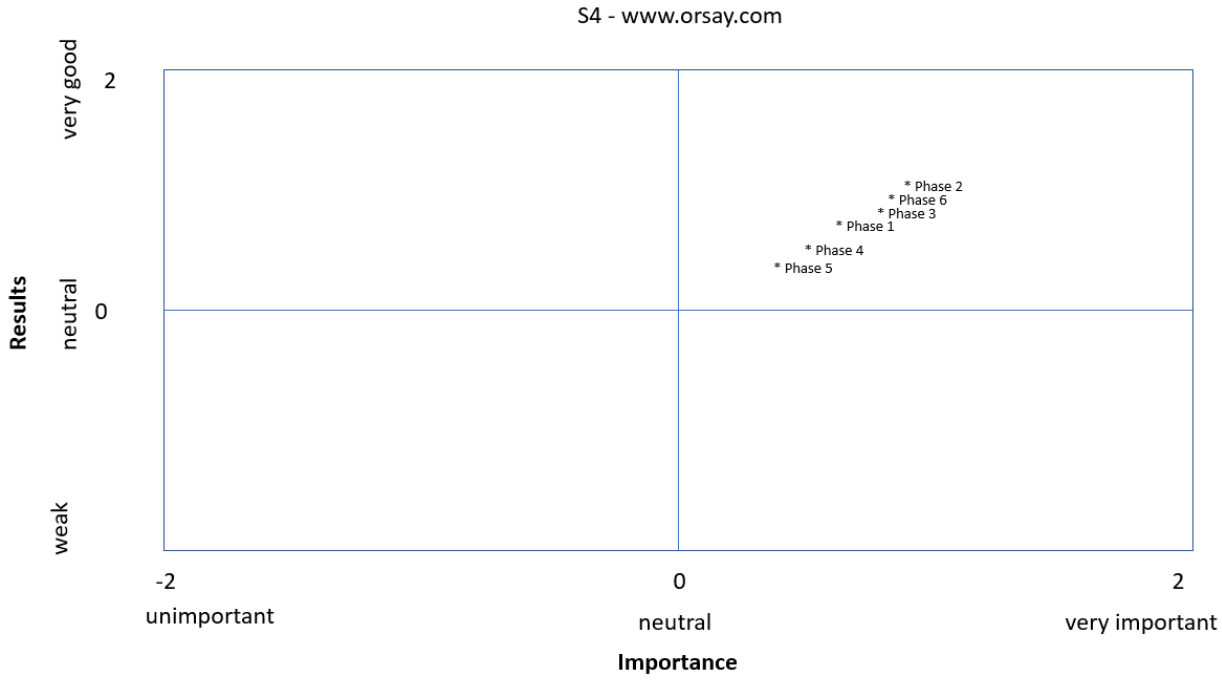


Figure 23 Graphical representation of S4 evaluation results

The second part of quantitative study consists of 10 questions that were performed to find if the EWAM methods criteria can be nowadays improved to evaluate e-commerce systems, especially in fashion sector, if the users' needs are changed due the modern

technology, internationalization/globalization, if the user needs are different – compare results from two Europe countries – Austria (central Europe) and Romania (eastern Europe). The question are optionally, not all the participants have answered.

The answers from the questions: “Does the site offer the option to view non-English pages? Does the site respect cultural differences between the home and foreign country? conclude that the e-commerce systems operate in selected target countries as well as it does in its home country, across geographic boundaries or technically performance – tables 17-18.

Table 17 Question 33 - answers results

			33. Does the site offer the option to view non-English pages?	CR	
			No	Yes - for all browsers	Yes - limited on browser preferences
www.zara.com	Austria	Female	0	2	1
www.zara.com	Austria	Male	1	0	7
www.zara.com	Romania	Female	0	1	1
www.zara.com	Romania	Male	0	0	1
www.mango.com	Austria	Female	0	1	10
www.mango.com	Austria	Male	0	0	0
www.mango.com	Romania	Female	0	0	0
www.mango.com	Romania	Male	0	0	0
www.hm.com	Austria	Female	1	0	1
www.hm.com	Austria	Male	0	1	8
www.hm.com	Romania	Female	0	3	2
www.hm.com	Romania	Male	0	1	0
www.orsay.com	Austria	Female	0	1	9
www.orsay.com	Austria	Male	0	0	0
www.orsay.com	Romania	Female	0	0	1
www.orsay.com	Romania	Male	0	0	0
			2	10	41

Table 18 Question 34 - answers results

			34. Does the site respect cultural differences between the home and	CS
			No	Yes
www.zara.com	Austria	Female	1	2
www.zara.com	Austria	Male	1	7
www.zara.com	Romania	Female	1	1
www.zara.com	Romania	Male	1	0
www.mango.com	Austria	Female	3	8
www.mango.com	Austria	Male	0	0
www.mango.com	Romania	Female	0	0
www.mango.com	Romania	Male	0	0
www.hm.com	Austria	Female	2	0
www.hm.com	Austria	Male	2	7
www.hm.com	Romania	Female	2	3
www.hm.com	Romania	Male	0	1
www.orsay.com	Austria	Female	2	8
www.orsay.com	Austria	Male	0	0
www.orsay.com	Romania	Female	0	1
www.orsay.com	Romania	Male	0	0
			15	38

51 from 53 persons are answered that the site offers the option to view non-English pages but 15 from 53 are answered that the site does not respect cultural differences

between the home and foreign country. That is a good point with a potentially criteria to the method.

The answer to the questions: “Exists the possibility to select the products according to type of outfit? (e.g.: casual, elegant, clubbing, sport etc.) shows as good for the evaluated sites -table 19.

Table 19 Question 37 - answers results

			37. Exists the possibility to select the products according to type of outfit?	Cv
			No	Yes
www.zara.com	Austria	Female	2	1
www.zara.com	Austria	Male	1	7
www.zara.com	Romania	Female	0	2
www.zara.com	Romania	Male	0	1
www.mango.com	Austria	Female	5	6
www.mango.com	Austria	Male	0	0
www.mango.com	Romania	Female	0	0
www.mango.com	Romania	Male	0	0
www.hm.com	Austria	Female	2	0
www.hm.com	Austria	Male	3	6
www.hm.com	Romania	Female	2	3
www.hm.com	Romania	Male	0	1
www.orsay.com	Austria	Female	2	8
www.orsay.com	Austria	Male	0	0
www.orsay.com	Romania	Female	0	1
www.orsay.com	Romania	Male	0	0
			17	36

The answer to the questions: Exist a “complete the outfit” tool with complementary products to the current one? (Preferably chosen manually by a fashion editor) Has a commercial description done by fashion editor/specialist? (Emphasized as such – “what the specialist says”, “fashion editor’s recommendation” etc.) Is the presence of any of the following facilities: 360 degrees interactive pictures with the product, virtual fitting room or catwalk? show that this are a potentially improvement in fashion sector evaluations criteria and are shown in the following tables 20-22:

Table 20 Question 38 - answers results

			38. Exist a "complete the outfit" tool with complementary products to the	CW
			No	Yes
www.zara.com	Austria	Female	2	1
www.zara.com	Austria	Male	5	3
www.zara.com	Romania	Female	2	0
www.zara.com	Romania	Male	0	1
www.mango.com	Austria	Female	7	4
www.mango.com	Austria	Male	0	0
www.mango.com	Romania	Female	0	0
www.mango.com	Romania	Male	0	0
www.hm.com	Austria	Female	2	0
www.hm.com	Austria	Male	6	3
www.hm.com	Romania	Female	4	1
www.hm.com	Romania	Male	0	1
www.orsay.com	Austria	Female	6	4
www.orsay.com	Austria	Male	0	0
www.orsay.com	Romania	Female	0	1
www.orsay.com	Romania	Male	0	0
			34	19

Table 21 Question 39 - answers results

			39. Have a comercial description done by fashion editor/specialist?	CX
			No	Yes
www.zara.com	Austria	Female	3	0
www.zara.com	Austria	Male	7	1
www.zara.com	Romania	Female	2	0
www.zara.com	Romania	Male	0	1
www.mango.com	Austria	Female	11	0
www.mango.com	Austria	Male	0	0
www.mango.com	Romania	Female	0	0
www.mango.com	Romania	Male	0	0
www.hm.com	Austria	Female	2	0
www.hm.com	Austria	Male	9	0
www.hm.com	Romania	Female	3	2
www.hm.com	Romania	Male	0	1
www.orsay.com	Austria	Female	10	0
www.orsay.com	Austria	Male	0	0
www.orsay.com	Romania	Female	0	1
www.orsay.com	Romania	Male	0	0
			47	6

Table 22 Question 40 - answers results

			40. Is the presence of any of the following facilities: 360 degrees	CY
			No	Yes
www.zara.com	Austria	Female	3	0
www.zara.com	Austria	Male	8	0
www.zara.com	Romania	Female	1	1
www.zara.com	Romania	Male	1	0
www.mango.com	Austria	Female	11	0
www.mango.com	Austria	Male	0	0
www.mango.com	Romania	Female	0	0
www.mango.com	Romania	Male	0	0
www.hm.com	Austria	Female	2	0
www.hm.com	Austria	Male	9	0
www.hm.com	Romania	Female	4	1
www.hm.com	Romania	Male	1	0
www.orsay.com	Austria	Female	10	0
www.orsay.com	Austria	Male	0	0
www.orsay.com	Romania	Female	0	1
www.orsay.com	Romania	Male	0	0
			50	3

“Have you any other recommendation to improve the website evaluation? Please enter your recommendation” evidence that the particularly needs to for fashion e-commerce web sites users are changed due modern technology - virtual fitting room and catwalk are preferred – see figure 24.

42. Please enter your recommendation:	
virtual fitting room	12
catwalk	10
“complete the outfit” tool	5
360 degrees interactive pictures with the product	4
Security improovment	1

Figure 24 Question 42 - answers results

In the evaluation, it was found that the information phase is of the highest importance, and therefore the users of e-commerce sites in the fashion sector need new experiences like virtual fitting room or catwalk to complete their satisfaction.

5. FINAL CONCLUSION

The following conclusions can be drawn from the experimentation objectives:

- the applicability of the EWAM method and the assessment of e-commerce sites is viable. The results allow the global assessment of the quality of websites and the performance of comparative analyses between different websites.
- through the conducted experiments, the website evaluation criteria specified in the EWAM method were checked, but it cannot be said that they are validated. In order to validate the criteria, it is necessary to carry out new experiments that also take into account websites in other areas of applicability. At the same time, it is necessary to take into account the results published by other researchers in the field.
- the experiments carried out have a limited character due to the following main aspects:
  - the relatively small number of evaluators and the small number of sites analyzed (4);
  - failure to complete a full transaction that takes into account negotiation and contracting, as well as the technical assistance request from the supplier. Thus, only the potential to perform a particular transaction was considered in the experiment, without evaluating the actual performance of the transaction;
  - the results from experimentation have been compared with an international referential (“best practices”), which can lead to evaluations below the internationally accepted level in the field of e-commerce websites.
- overall, the results from the experiments are comparable to the results obtained by Schubert et al. (2003).

Based on the results of experimentation, the following proposals are formulated:

- developing “best practices” and “sector profile” referential to compare the quality of e-commerce applications with the average quality of a given sector and, at the same time, to compare best practices used in the sector concerned;
- performing experiments to evaluate a sufficient number of e-commerce applications in a particular sector, and compare the results with client satisfaction criteria;
- developing recommendations or technical specifications on the quality of e-commerce applications.

In the evaluation, it was found that the information phase is of the highest importance, and therefore the new end-user requirements can be formulated as criteria in assessing e-commerce sites in the fashion sector - in particular. Therefore, the H1 hypothesis should respond affirmatively. This result also confirms that new criteria may develop for the future due to the development of modern technology and internationalization / globalization to improve the evaluation of e commerce web-sites.

Appendix 4.3

The significance of the rating given and the ideal situation of the criteria

Information Phase
EOU01 – Website accessibility and product offerings Question: Is the website easy to find by users – potential clients of the products they offer?

Criterion importance evaluation	Significance
++	Very easy to find offer, very easy to remember site address
+	Easy to find offer, easy to remember site address
-	Difficult to find offer, difficult to remember site address
--	Very difficult to find offer, very difficult to remember site address

Ideal situation:

- The Internet address (URL) is easy to remember and easy to insert.
- The offer is recognized by major search engines and software agents (the use of metatags is recommended to adequately describe the content of the website to anticipate potential search criteria for prospective users – clients).
- "Pull" mechanism: a supplier's offer is linked to other WWW areas in which the target group circulates; this implies the supplier's knowledge of the exact profile of the target market segment.
- The Internet offer is based on a strong, well-known brand (users can specify it directly in a search with a browser).

EOU02 – Content structure

Question: How do you rate the structure of the content – the presentation mode and the logic of the website's construction?

Criterion importance evaluation	Significance
++	The content is very well structured
+	The content is well structured
-	The content is poorly structured
--	The content is very poorly structured

Ideal situation:

- Visitors find the content arranged from their individual perspective.
- Special clients find *New Products*, *Offers*, and *Special Conditions* on the website.
- Navigation help allows easy access to content (e.g., products are arranged on categories by product groups, prices, destinations, etc.).
- The basic information important for the visitor is well placed and can be easily found immediately, and further detailed information can be quickly obtained.
- The search functions provided within the site act quickly and effectively.
- Company information (organization, address, contact points, etc.) is easily accessible.

EOU03 – Amount of information

Question: How do you assess the amount of information made available to describe the products and services offered (relevance and quantitative dosing)?

Criterion importance evaluation	Significance
++	The amount of information is very well dosed
+	The amount of information is well dosed
-	Pages are partially overloaded or information is occasionally missing
--	Pages are usually overloaded or there is a general lack of information

Ideal situation:

- Visitors receive the information they need without being overloaded with information.

- The company and its products and services are described at a sufficient level of detail.
- The amount of information is so dosed that the visitor is not lost in detail and is able to achieve his/her goal.

USEF01 – Quality of content

Question: Does the quality of the website's content meet client expectations?

Criterion importance evaluation	Significance
++	Content quality is very good
+	Content quality is good
-	Content quality is poor
--	Content quality is bad

Ideal situation:

- The content is of high quality and presents the correct and constantly updated description of the company and its products and services.
- Descriptions are accompanied by photos and images.
- The content contains additional important information compared to other sources.
- Price comparisons with competing bids are presented.

USEF02 – Transfer by cost of the supplier's cost-related benefits

Question: Are the benefits of the supplier due to the cost reduction found in the lower prices offered to the client?

Criterion importance evaluation	Significance
++	Prices are significantly lower than in a classic store
+	Prices are slightly lower than in a classic store
-	Cost benefits are not reflected in client prices

--	Prices are significantly higher than in a classic store

Ideal situation:

- The relative cost benefits compared to classic stores are transferred to the client in the form of lower prices.
- Postage costs do not lead to a more expensive online purchase than in a classic store.

USEF03 – Packages of products and services

Question: Are products and services (possibly provided by a third party) tied together to meet client requirements, accompanied by an appropriate description? Is a client help for configuring complex products and choosing complementary services offered by other providers made available? Are prices and offers for combinations of special products presented to facilitate the client's decision? Are undesirable or impossible product combinations signaled?

Criterion importance evaluation	Significance
++	Very good presentation of product and service packages
+	Good presentation of product and service packages
-	Low presence of product and service packages
--	Lack of presentation of product and service packages, although desirable

Ideal situation:

- The offer includes complementary products and services.
- The offer is supplemented by services provided by third parties.
- The prices and conditions of the integrated third-party supplier are described comprehensibly.
- Clients can configure their products or services as desired.
- Presentation of product configurations, checklists, planning systems, etc. to facilitate the choice of products and components.

USEF04 – Recommendation systems

Question: How well do the recommendation systems from the website (automatically generated by the profile of the buyer or by filtering similar community preferences) function?

Criterion importance evaluation	Significance
++	Very good recommendation systems; the quality of the recommendations increases with the size of the community and the intensity of client relationships
+	Good recommendation systems
-	Recommendation systems present
--	There are no recommendation systems, although desirable

Ideal situation:

- Pre- and inter-sale tools make complementary proposals for the base product.
- Collaborative filtering systems offer product recommendations that meet client expectations.
- The quality of the recommendations increases with the size of the community and the intensity of the relationships.

USEF05 – Hypermedia usage

Question: How does hypermedia usage affect you?

Criterion importance evaluation	Significance
++	Very good use of hypermedia
+	Good use of hypermedia
-	Unsatisfactory use of hypermedia
--	Insufficient use of hypermedia

Ideal situation:

- Quality: the hypermedia used on the website is very well done technically.
- Functionality: the hypermedia used gives the user an extra real benefit.
- Adequate: the hypermedia variety used covers the needs of the user.

Negotiation and Contracting Phase

EOU04 – Product and service order procedure design

Question: Is the product and service order procedure simple and clear (from product selection, price finding, selection of the payment method, and choice of delivery)?

Criterion importance evaluation	Significance
++	The ordering of products and services is very simple and very clear
+	The ordering of products and services is simple and clear
-	The ordering of products and services is complicated and unclear
--	The ordering of products and services is very complicated and too unclear

Ideal situation:

- Each step in the order procedure is explained to the client in a precise and easy to understand manner.
- Clients know at any time the stage of the procedure and what the consequences of the next step are, with an easy way of returning to the previous step or dropping the order at any time.
- For relevant contract data (e.g., general sales conditions, payment methods, pricing), there are always *accept buttons* that need to be activated.
- After the first order, the procedure is shortened if the client does not change the payment method and the delivery form at each new purchase.
- Negotiating the price is as easy and intelligible as possible. The same goes for potential service providers.
- The provider offers several ways to order from which clients choose the preferred method (order by e-mail, online order form, shopping cart).

USEF06 – Models and pricing methods

Question: Are the applied models and available pricing methods determining fair and individualized prices?

Criterion importance evaluation	Significance
++	Fair, individual and convenient, client-dependent prices
+	Prices considered convenient
-	Prices considered inconvenient
--	General, incorrect prices

Ideal situation:

- The pricing methods offered are appropriate to the transaction and result in a correct individualized price (depending on buy frequency, quantity and value of purchases, discount system, etc.).
- The pricing model is appropriate for the transaction (fixed prices for immediate delivery, negotiated prices, etc.).
- Loyal clients benefit from individual price reductions depending on the client profile.

Acquisition and Payment Phase

EOU05 – Integration of generic services

Question: Is it easy to choose in the order procedure between the generic services for the payment and logistics procedure?

Criterion importance evaluation	Significance
++	Generic services are well integrated and can be very easily selected
+	Generic services are integrated and can be easily selected
-	Generic services can only be selected with difficulty

--	Generic services are not integrated, though desirable

Ideal situation:

- Clients can pay more easily by choosing the preferred payment method: credit card (Visa, Mastercard, etc.), pro-forma invoice, check, payment order, mobile phone payment, cash.
- Simple and individual choice of your preferred distribution partner (by mail, courier, from supplier warehouse, etc.) and delivery date.
- All terms and conditions resulting from the choice of distribution channel, date of delivery and payment modalities are documented in an intelligible manner.).

USEF08 – Order traceability

Question: Can the order's status be tracked?

Criterion importance evaluation	Significance
++	Very good traceability services
+	Good traceability services
-	Unsatisfactory traceability services
--	Inexistent traceability services, although desirable

Ideal situation:

- Online services available for tracking each step of an order from order placement to goods delivery to the recipient and back after receiving the goods.
- Individual client order data is easily accessible, secure, and understandable (enabling the client to learn the status of the order and the status of the corresponding deliveries).

After-Sales Service Phase

EOU06 – Access to client support

Question: Is it easy to use client support present on the website (contact details, running hours, access channels, establishing physical contact with a company personally)?

Criterion importance evaluation	Significance
++	Client support is very easy to access
+	Client support is easy to access
-	Client support is difficult to access
--	It is not possible to access the client support, even desirable

Ideal situation:

- Client support is easily accessible on the website.
- Client support is available 24/7.
- Client support is accessible by e-mail, telephone and letter / fax.
- Client support uses the same environment as the callback (reply to e-mail, etc.).

EOU09 – Client support utility

Question: How satisfied are you with the quality of the client support present on the website (answers to questions, additional information for client education – in the form of online presentation materials, error correction help, online libraries, etc., – and procedures which solve problems after buying a product – return, complaints and possible remedies)?

Criterion importance evaluation	Significance
++	Very helpful client support
+	Helpful client support
-	Poor client support
--	Inexistent client support

Ideal situation:

- Client support responds quickly and securely to client requests.
- Clients receive all necessary information regarding the use of the purchased product.
- There is an individual relationship between the buyer and client support. Clients have the feeling that their requests are solved personally.
- Clients receive unsolicited information (as long as they agree) related to new products and updates (e.g., in the form of a customized newsletter).

Community Component

EOU07 – Access to a virtual community

Question: Is there access to a virtual community and is it easy to find?

Criterion importance evaluation	Significance
++	The virtual community is very easy and fast to access
+	The virtual community is easy to access
-	The virtual community is difficult to access
--	It is not possible to access the virtual community, though desirable

Ideal situation:

- The website provides access to a relevant virtual community.
- The relevant virtual community is integrated directly into the offer, and is therefore easy to find.

USEF10 – Usefulness of relationships in the virtual community

Question: How is the quantity and quality of relationships within the virtual community (forum participation, etc.) evaluated as a support for the selection and purchase of products and services?

Criterion importance evaluation	Significance
++	Very good relationships
+	Good relationships
-	Weak relationships
--	Bad relationships

Ideal situation:

- The community has a sufficient number of members registered.
- There is a network of trusted relationships within the community.
- The community offers the opportunity to get in contact with experts and other specialists otherwise difficult to approach.
- Members of the community treat each other with respect, feel that they belong to it.
- Community behavior rules are clearly stipulated and supervised by either the community's organizer or by the user's own organization (*netiquette*).

USEF11 – Usefulness of content accessible through the virtual community

Question: How is content accessed through the virtual community (as a source and filter for customized presentation for a particular client) evaluated?

Criterion importance evaluation	Significance
++	Very good content in the virtual community
+	Good content in the virtual community
-	Insufficient content in the virtual community
--	Bad content in the virtual community

Ideal situation:

- The main community orientation reflects the interests of the buyer group (the target group).
- The community is a specialized knowledge repository, knowledge that can be obtained by a member only from here.

- Members of the community generate – apart from officially published content – their own valuable contributions to the community.
- The community provides truthful and understandable information from a wide range of sources.
- There is an adequate level of exchange of information between community members.

USEF12 – Power of client influence in the virtual community

Question: Does the virtual community generate a client's power of influence on the supplier (through the power of more clients and market transparency) and bring benefits to the client member of the community?

Criterion importance evaluation	Significance
++	The virtual community generates a strong power of influence on the client
+	The virtual community generates a power of influence on the client
-	The virtual community does not generate power of influence on the client
--	The virtual community determines higher prices

Ideal situation:

- The virtual community has the role of consumer protector.
- The community leads to greater market transparency and eliminates the inconveniences and scammers.
- Buyer communities obtain discounts in quantitative criteria.

Final Section

EOU08 – System availability

Question: Is the website available and functional anytime without interruption (daily, 24/7)?

Criterion importance evaluation	Significance
++	The website is permanently available
+	Certain pages are sometimes unavailable
-	Certain pages are often unavailable
--	The entire website is often unavailable

Ideal situation:

- Requesting information and ordering products and services can be made at any time.
- The product and service orders in the offer can be launched at any time of the day and in all day of the week.

EOU09 – User interface design

Question: How do you appreciate the site's user interface (consistency, standardized form, ease of use, target user group addressability)?

Criterion importance evaluation	Significance
++	Very good user interface
+	Good user interface
-	Poor user interface
--	Bad user interface

- Ideal situation:
- The interface is easy to use.
- Interface design is standardized and consistent.
- Navigation is easy and intuitive.
- Interface design is devised so as to meet the needs of a visitor who accesses the website for the first time, determining him/her to go through the pages of the website.
- Help functions are permanently available and are easy to use.
- Interface components should not trigger any compatibility issue for the user.

- Global aspects were considered when designing the user interface (language used for user groups, texts according to local and cultural specifics, correct (grammatically, syntactically, content-wise).

USEF13 – Productivity increase by reducing the time consumed

Question: Can you use the website to reduce client time? The reduction is given not only by the time required to go to the classic store or make a phone call, but also by the interaction with the website (how fast the website loads, a transaction is made, and the ordered product or service is delivered).

Criterion importance evaluation	Significance
++	Great reduction of time consumed
+	Insignificant reduction of time consumed
-	No reduction of time consumed
--	Waste of time

Ideal situation:

- Compared to traditional (store or phone) purchases and other websites, the website allows clients to reduce their time spent on all phases of the transaction.
- The load and response times of the website are fast or at least acceptable.

USEF14 – Interaction

Question: Can the website be considered interactive (allows direct contact with the client, responds to his/her needs, and influences every phase of the transaction process)?

Criterion importance evaluation	Significance
++	Very good interaction
+	Good interaction
-	Too much or too little interaction
--	Far too much or far too little interaction

Ideal situation:

- The website is designed for interaction; clients can choose according to their own needs from different procedural options.
- For products requiring more information, an online chatroom consultant or an online configurator is available to configure interactively the products.

USEF15 – Customization features

Question: Does the website respond to individual needs or provide the appropriate information for the client's profile (customized)?

Customization is based on the information requested and filled in by the client or derived from his/her behavior and stored in the client profile?

Criterion importance evaluation	Significance
++	Very good customization features
+	Good customization features
-	Poor customization features
--	Inexistent customization features, though desirable

Ideal situation:

- The client can log in personally to the website.
- The content of the website may be partially chosen and arranged by the client.
- When reviewing the website, clients do not have to provide detailed information that was already completed (address, form of payment, preferences, etc.).
- Clients have access to their previous orders (history).
- Clients receive recommendations on the products and accessories in the offer based on their previous purchases and / or their preferences.
- All personalization features lead to a reduction in the length of the decision making procedures.

TRUST01 – Business partner (supplier) trust

Question: Does the business partner (supplier) inspire confidence?

Criterion importance evaluation	Significance
++	The business partner inspires great confidence
+	The business partner inspires confidence
-	The business partner inspires little confidence
--	The business partner does not inspire confidence at all

Ideal situation:

- The company is well presented and described on the website (in the *About Us* section).
- The manager and the team directly involved in the relationship with the client are presented on the website with their photos, including contact opportunities.
- The terms of sale and, in particular, the means of complaining and returning goods are client-friendly and comprehensibly written.
- Clients need to give only essential information, other details are voluntary.
- The website provides access to a community where the provider and products are evaluated.
- Existence of proof of supplier's trust on the website (references, awards, etc.).
- The company assumes responsibility for maintaining the confidentiality of client data.

TRUST02 – Website respectability and legal situation

Question: The website and the legal situation (the legal framework of online transactions) inspire client trust in buying a product (site free from defects, ensuring the not allowed non-use of personal data, safety against hacker attacks or viruses)?

Criterion importance evaluation	Significance
++	The website inspires great confidence
+	The website inspires confidence
-	The website inspires little confidence

--	The website does not inspire confidence

Ideal situation:

- Provision of safety precautions and their description.
- The website is certified.
- Clients are informed of legal requirements related to the protection of personal data and ensuring compliance with them.
- The sales conditions and legal aspects of the supplier’s country of origin are described by everyone.

Appendix 2.2

One of the first models for measuring quality of software can be found in McCall, Richard & Walters, in which the authors identify 55 candidates for quality factors, these reduce the assessment procedures to finally eleven quality factors are presented below:

- Correctness
  "Extend to which a program satisfies is specifications and fulfils the user's mission objectives" (McCall, Richard & Walters, 1978)
- Reliability
  "Extend to which a program can be expected to perform it's intended function with required precision" (McCall, Richard & Walters, 1978)
- Efficiency
  "The amount of computing resources and code required by a program to perform a function" (McCall, Richard & Walters, 1978)
- Integrity
  "Extend to which access to software or data by unauthorized person can be controlled" (McCall, Richard & Walters, 1978)
- Usability
  "Effort required learning, operating, preparing input and interpreting output of a program" (McCall, Richard & Walters, 1978)
- Maintainability
  "Effort required to locate and fix an error in an operational program" (McCall, Richard & Walters, 1978)
- Testability
  "Effort required to test a program to insure its intended function" (McCall, Richard & Walters, 1978)
- Flexibility
  "Effort required to modify an operational program" (McCall, Richard & Walters, 1978)
- Portability
  "Effort required to transfer program from one hardware configuration and/or software system environment to another" (McCall, Richard & Walters, 1978)
- Reusability
  "Extend to which a program can be used in another applications – related to the packaging and scope of the functions that programs perform" (McCall, Richard & Walters, 1978)
- Interoperability
  "Effort required to couple one system with another" (McCall, Richard & Walters, 1978)

6. BIBLIOGRAPHY

Böehm, R.W., Brown, J.R., Kaspar, H., Lipow, M., Macleod, G.J.&Merrit, M.J (1978) *Charcteristics of software quality,* Amsterdam:Elsevier

Balog, A. *The quality of interactive systems – studies and experiments,* Bucuresti, Matrix Rom 2004

Hoffmann, W.D. (2008). *Software-Qualität*. Berlin: Springer.

IEEE Computer Society. (2008). IEEE Standard for Software and System Test Documentation. New York

ISTQB International Software Testing Qualifications Board (2011)**,** *Foundation Level Syllabus*

Kan, Stephen H. (2003). Metrics and Models in Software Quality Engineering. Addison-Wessley

McCall, J.A., Reichards, P.K. & Walters, G.F. (19978). *Factors in software quality*

Schubert, P. *Extended Web Assessment Method (EWAM) – Evaluation of Electronic Commerce Applications from the customer's Viewpoint,* in Proceddings of the 35th HICSS Conference, Hawaii, 2002

Schubert, P. *Extended Web Assessment Method (EWAM) – Evaluation of Electronic Commerce Applications from the customer's Viewpoint,* in International Journal of Electronic Commerce, Vol. 7, No. 2, Winter 2002-2003

Spillner, A & Linz, T. (2004). *Basiswissen Softwaretest*. Heidelberg: dpunkt.verlag

Figure Contents

Tables Contents