

The COCA quality model for user documentation

Bartosz Alchimowicz · Jerzy R. Nawrocki

Published online: 12 October 2014

© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract According to Pedraz-Delhaes, users evaluate both the product and the vendor on the basis of provided documentation. Thus, a question arises as to what quality characteristics should be taken into account when making a decision about accepting a given user manual. There are some proposals (e.g., ISO Std. 26513 and 26514), but they contain too many quality characteristics and lack orthogonality. The goal of this paper is to propose a simple quality model for user documentation, along with acceptance methods based on it. The model is to be orthogonal and complete. As a result, the COCA quality model is presented, which comprises four orthogonal quality characteristics: Completeness, Operability, Correctness, and Appearance. To check completeness, the proposed quality model has been compared with many other quality models that are directly or indirectly concerned with user documentation. Moreover, two acceptance methods are described in the paper: pure review based on ISO Std. 1028:2008, and documentation evaluation test (a type of browser evaluation test), which is aimed at assessing the operability of user documentation. Initial quality profiles have been empirically collected for both methods—they can be used when interpreting evaluation results obtained for a given user manual.

Keywords User documentation · Quality model · Systematic evaluation · Documentation evaluation test

1 Introduction

A good quality user manual can be beneficial for both vendors and users. According to Fisher (2001), a project can be called successful if its *software performs as intended and the users are satisfied*. From the point of view of end users, the intended behavior of a

B. Alchimowicz (✉) · J. R. Nawrocki
Institute of Computing Science, Poznań University of Technology, ul. Piotrowo 2, 60-965 Poznań,
Poland
e-mail: bartosz.alchimowicz@cs.put.poznan.pl

J. R. Nawrocki
e-mail: jerzy.nawrocki@put.poznan.pl

software system is described in the user manual. Thus, a defective user manual (e.g., lack of consistency with the software system) has an effect similar to defective software (off specification)—both will lead to user irritation, which will decrease user satisfaction. Pedraz-Delhaes et al. (2010) also point out that users evaluate both the product and the vendor on the basis of provided documentation. According to the data presented by Spencer (1995), a good quality user manual can reduce the number of calls from 641 to 59 over a 5-month period (in 2008, the average cost of support for one call was above \$32 (Markel 2012)).

Unfortunately, end users are too frequently dissatisfied with the quality of their user manuals. They complain that the language is too hard to understand, the descriptions are boring, and the included information is outdated and useless (Novick and Ward 2006a, b). Some users even feel frustrated while working with the software (Hazlett 2003).

So, a good quality user manual is important. Thus, the question arises of what *good quality* means in this context, i.e., what quality characteristics should be considered when evaluating the quality of a user manual. A set of quality characteristics constitutes a quality model (ISO/IEC 2005), and these should be *orthogonal* (i.e., there should be no overlap between any two characteristics) and *complete* (i.e., all the quality aspects important from a given point of view should be covered by those characteristics).

In this paper, an orthogonal and complete quality model for user documentation is presented. The model is called COCA and consists of four quality characteristics: Completeness, Operability, Correctness, and Appearance. From the practical point of view, what matters is not only quality characteristics, but also the way they are used in the evaluation process. As indicated by the requirements of Level 4 of Documentation Maturity Model (Huang and Tilley 2003), quality characteristics should allow quantitative assessment. In this paper, two approaches are discussed, a review-based evaluation and an empirical one. Both of them provide quantitative data. For each of them, quality profiles for the educational domain are presented, which can be used when interpreting evaluation data obtained for a particular user documentation.

The paper is organized as follows: In Sect. 2, a set of design assumptions for the proposed quality model is presented. Section 3 contains the COCA quality model. Section 4 shows how the proposed model can be used. Section 5 presents an empirical approach to operability assessment. Related work is discussed in Sect. 6. A summary of the findings and conclusions are contained in Sect. 7.

2 Design assumptions for the quality model

As defined by ISO Std. 25000:2005, a quality model is a *set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality*.

The quality model described in this paper is oriented toward user documentation, understood as *documentation for users of a system, including a system description and procedures for using the system to obtain desired results* (ISO/IEC/IEEE 2010).

The design assumptions for the quality model are presented in the subsequent parts of this section.

2.1 Form of user documentation

User documentation can have different forms. It can be a PDF-like file ready to print, a printed book, on-screen information or standalone online help (ISO/IEC/IEEE 2011).

Assumption 1 It is assumed that user documentation is presented in the form of a static PDF-like file.

Justification On-screen help is based on special software, and to assess its quality, one would have to take into account the quality characteristics appropriate for the software, such as those presented in one of the ISO standards (ISO/IEC 2011). That would complicate the quality model, and the aspects which are really important for user documentation would be embedded into many other characteristics. Thus, for the sake of clarity, such forms of user documentation as on-screen help are out of the scope of the presented model. To be more precise, on-screen help can be evaluated on the basis of the proposed model, but to have a complete picture, one should also evaluate it from the software point of view. □

2.2 Point of view

The quality of user documentation can be assessed from different points of view. Standards concerning user documentation presented by ISO describe a number of roles that are involved in the production and usage of user documentation (e.g., suppliers (ISO/IEC/IEEE 2011)), testers and reviewers (ISO/IEC 2009), designers and developers (ISO/IEC 2008), and users for whom such documentation is created).

Assumption 2 It is assumed that user documentation is assessed from the end users' point of view.

Justification People may have different requirements for user documentation, and thus, they focus on different aspects, i.e., project managers may want to have documentation on time, while designers may be interested in creating a pleasing layout. However, all work that is done aims to provide user documentation that is satisfactory for end users. Thus, their perspective seems to be the most important. As a consequence, legal aspects, conformance with documentation design plans, etc., are neglected in the proposed model. □

2.3 External quality and quality-in-use

The software quality model presented in ISO/IEC Std. 9126:1991 was threefold: the internal quality model, the external quality model, and the quality-in-use model. From the users' point of view, internal quality seems negligible and as such is omitted in this paper. We are also not taking into account the relationship between user documentation and other actors, such as the documentation writer. Considering the above, the following assumption seems justified:

Assumption 3 A quality model for user documentation can be restricted to characteristics concerning external quality and quality-in-use.

2.4 Context of use

There are many possible contexts of use for user documentation. One could expect that such documentation would explain scientific bases of given software or compare the software against its competitors. Although this information can be valuable in some contexts, it seems that text books or papers in professional journals would be more appropriate for this type of information. Thus, the following assumption has been made when working on the proposed quality model:

Assumption 4 User documentation is intended to support users in performing business tasks.

2.5 Orthogonality of a quality model

Definition 1 A quality model is orthogonal, if for each pair of characteristics C_1, C_2 belonging to it, there are objects O_1, O_2 which are subject to evaluation such that O_1 gets a highly positive score with C_1 and a highly negative score with C_2 , and for O_2 it is the opposite. \square

Assumption 5 A good quality model for user documentation should be orthogonal.

Justification If a quality model is not orthogonal, then it is quite possible that some of its characteristics are superfluous, as what they show (i.e., the information they bring) can be derived from the other characteristics. For instance, when considering the sub-characteristics of ISO Std. 9126 (ISO/IEC 2001), one may doubt whether changeability and stability are orthogonal, as one strongly correlates with the other (see Jung et al. 2004). \square

2.6 Completeness of a quality model

The completeness of a quality model should be considered in the context of the point of view of a stakeholder. This point of view can be characterized with the set of *quality aspects* one is interested in. A *quality aspect* is a type of detailed information about quality. Using terminology from ISO Std. 9126 and ISO Std. 25010 (ISO/IEC 2011), a *quality aspect* could be a quality sub-characteristic, sub-subcharacteristic, etc. An example of a *quality aspect* could be completeness of documentation from the legal point of view (that could be important from a company standpoint) or the presence of a table of contents. Many *quality aspects* can be found in standards such as ISO Std. 26513 and ISO Std. 26514 (ISO/IEC 2008, 2009).

Definition 2 A quality model is complete from a given point of view, if every quality aspect important from that point of view can be clearly assigned to one of the quality characteristics belonging to the quality model. \square

Assumption 6 A good quality model for user documentation should be complete from the end user point of view.

The above assumption follows from Assumption 2.

3 The COCA quality model

The COCA quality model presents the end users' point of view on the quality of user documentation. As its name suggests, it consists of four quality characteristics: Completeness, Operability, Correctness, and Appearance. Those characteristics are defined below.

Definition 3 Completeness is the degree to which user documentation provides all the information needed by end users to use the described software. \square

Definition 4 Operability *sensu stricto* (Operability for short) is the degree to which user documentation has attributes that make it easy to use and helpful when acquiring information that is contained in the user documentation. □

Justification There are two possible definitions of Operability: *sensu stricto* and *sensu largo*. Operability *sensu largo* could be defined as follows:

Operability *sensu largo* is the degree to which user documentation has attributes that make it easy to use and helpful when operating the software documented by it.

Operability *sensu largo* depends on two other criteria: Completeness and Correctness. If some information is missing from a given user manual or it is incorrect, then the helpfulness of that user manual is diminished when operating the software. Operability *sensu largo* is not a characteristic of a user manual itself, but is also depends on (the version of) the software. For instance, Operability *sensu largo* of a user manual can be high for one version of software, and low for another, newer version, if that new version of software was substantially extended with new features. Thus, Operability *sensu largo* is not orthogonal with Completeness and Correctness. Operability *sensu stricto* is defined in such a way that it is independent of Completeness or Correctness of the user manual. It depends only on the way in which a user manual is made up and how it is organized. To preserve orthogonality of the proposed quality model, Operability *sensu stricto* has been chosen over Operability *sensu largo*. □

Definition 5 Correctness is the degree to which the descriptions provided by the user documentation are correct. □

Definition 6 Appearance is the degree to which information contained in user documentation is presented in an aesthetic way. □

As mentioned earlier, it is expected that the COCA quality model is both orthogonal and complete. These issues are discussed below.

Claim 1 *The COCA quality model is considered orthogonal.*

Justification Since the COCA quality model consists of four characteristics, one has to consider 6 pairs of them. All of the pairs are examined below, and, for each of them, two manuals which would lead to opposing evaluations are described.

Completeness versus Operability

When a user manual contains all the information, a user needs to operate a given software, but the user manual is thick and ill-designed (no index, exceedingly brief table of contents, all text formatted with a single font type without underlining, etc.), then such a user manual would be highly complete, but its operability would be low. And vice versa: a user manual can be highly operable (i.e., its Operability *sensu stricto* can be high) but still be missing a lot of important information, causing its completeness to be low. That shows that Completeness and Operability are orthogonal.

Completeness versus Correctness

It is possible that a user manual covers all the aspects concerning usage of a given software, but the screen shots still refer to the old version of the software. Similarly,

business logic described in the user manual may be based on outdated law regulations, etc., which meanwhile have been changed in both the real world and in the software, but not in the user manual. And the contrary is also possible: All the descriptions provided by a user manual can be correct, but some important information can be missing (e.g., about new features added to the software recently). Thus, Completeness and Correctness are orthogonal.

Completeness versus Appearance

It is pretty obvious that a document can be highly complete, as far as information is concerned, but far from giving an impression of beauty, a good taste, etc., and vice versa. Therefore, Completeness and Appearance are orthogonal.

Operability versus Correctness

According to Definition 4, Operability is the degree of ease of finding information contained in the user manual. It does not take into account whether or not that information is correct. Because of this, Operability and Correctness are orthogonal.

Operability versus Appearance

According to Definition 6, Appearance is about aesthetics. According to the Free Dictionary¹, aesthetics is about *beauty or good taste*. Here are several examples of factors that can impact the aesthetics of a user manual:

- the chosen set of font types (many different font types can increase Operability, but decrease aesthetics; small font types can increase aesthetics but decrease Operability);
- the set of colors used in the document (red and green can increase Operability but, if used improperly, can decrease the aesthetic value of a user manual);
- screenshots (they can be very valuable from the Operability point of view, but—if not properly placed—can decrease the aesthetics of a user document);
- decorative background (though favoured by some, it can decrease the readability of a document; thus, it can decrease its Operability).

These factors can create a trade-off between the aesthetics and Operability of a user manual; thus, Operability and Appearance can be regarded as orthogonal.

Correctness versus Appearance

It seems pretty clear that those two characteristics are orthogonal; a document can be highly correct but its Appearance can be low, and vice versa. □

Claim 2 *The COCA quality model is considered complete.*

Justification To check completeness of the COCA model, the model will be examined from the point of view of the following sets of quality characteristics: ISO Std. 26513 and ISO Std. 26514 (ISO/IEC 2008, 2009), Markel's measures of excellence (Markel 2012), Allwood's characteristics (Allwood and Kalén 1997), Ortega's systemic model (Ortega et al. 2003), and Steidl's quality characteristics for comments in code (Steidl et al. 2013).

If talking about completeness, it is important to distinguish between two notions:

¹ <http://www.thefreedictionary.com/aesthetic>

- documentation-wide *quality aspects*: all of them should be covered by a quality model if that model is to be considered complete;
- documentation *themes*: all of them should be covered by a user manual if that manual is to be considered complete.

Here are the documentation *themes* identified on the basis of ISO Std. 26513 and ISO Std. 26514:

- description of warnings and cautions,
- information about the product from the point of view of appropriateness recognizability,
- information on how to use the documentation,
- description of functionality,
- information about installation (or getting started).

If one of those *themes* is missing, the documentation can be incomplete in the eye of an end user. Thus, documentation *themes* influence *Completeness* of a user manual, but do not directly contribute to a quality model.

The *quality aspects* that can be found in ISO Std. 26513 and ISO Std. 26514 are listed in Table 1. They can be mapped into the three COCA characteristics: *Operability* (covers ease of understanding and consistency of terminology), *Correctness* (it corresponds to consistency with the product), and *Appearance* (it is influenced by consistency with style guidelines, editorial consistency, and cultural requirements). Thus, from the point of ISO Std. 26513 and ISO Std. 26514, the COCA model seems complete.

Completeness of the COCA quality model can be also examined against Markel's model of quality of technical communication (Markel 2012). Merkel's model is based on eight measures of excellence. Seven of them are presented in Table 2 and they are covered by the COCA characteristics. The eighth measure of excellence is *honesty*. It does not fit any of the COCA characteristics. However, it is not an external quality nor a quality-in-use characteristic, so—according to Assumption 3—it is out of scope of the defined interest. Thus, the COCA model, when compared against Markel's measures of excellence, is considered complete.

Another set of quality characteristics has been presented by Allwood and Kalén (1997). Two of them, i.e., *comprehensibility* and *readability*, are covered by COCA's *Operability* (if a document lacks *comprehensiveness* or *readability* then acquiring information from it is difficult, so COCA's *Operability* will be low). The third Allwood's characteristic is *usability*. It is a very general characteristic, which is influenced by both *comprehensibility* and *readability*. When comparing it to the COCA characteristics, one can find that *usability* encompasses COCA's *Completeness*, *Operability*, and *Correctness*, i.e., Allwood's *usability*

Table 1 Documentation-wide *quality aspects* versus COCA characteristics

Quality aspect (ISO Std. 26513 and ISO Std. 26514)	COCA characteristics
Ease of understanding Consistency of terminology	Operability
Consistency with the product	Correctness
Consistency with style guidelines Editorial consistency Cultural requirements	Appearance

Table 2 Markel's measures of excellence (Markel 2012) versus COCA characteristics

Markel's measures of excellence	COCA characteristics
Comprehensiveness <i>A good technical document provides all the information readers need</i>	Completeness
Clarity <i>Your goal is to produce a document that conveys a single meaning the reader can understand easily</i>	Operability
Accessibility <i>Readers should not be forced to flip through the pages . . . to find the appropriate section</i>	
Conciseness <i>A document must be concise enough to be useful to a busy reader</i>	
Accuracy <i>A major inaccuracy can be dangerous and expensive</i>	Correctness
Professional appearance <i>Document looks neat and professional</i>	Appearance
Correctness <i>A correct document is one that adheres to the conventions of grammar, punctuation, spelling, mechanics, and usage</i>	

Table 3 Ortega's quality characteristics (Ortega et al. 2003) versus COCA characteristics

Ortega's characteristics	COCA characteristics
Completeness	Completeness
Learnability	Operability
Self-descriptiveness	
Understandability	
Consistency	Correctness
Attractiveness	Appearance

can be regarded as a triplet of COCA's characteristics. Allwood also mentioned two other quality characteristics: *interesting* and *stimulating*. As we are interested in user documentation as support in performing business tasks (see Assumption 6), those characteristics can be neglected. Thus, one can assume that the COCA model is complete in its context of use.

Other quality characteristics the COCA model can be examined against are Ortega's characteristics (Ortega et al. 2003). Although those characteristics are oriented toward software products, they can be translated into the needs of user documentation, see Table 3. For instance, *learnability*, in the context of user documentation, can be understood as the degree to which it is easy to learn how to use a given user documentation. So,

learnability is part of COCA's *Operability*. Similar meaning can be given to *self-descriptiveness* in the context of user documentation. Ortega's *understandability* also fits COCA's *Operability*, as it supports acquiring information from documentation. *Consistency* of software can be translated into consistency of user documentation with its software, so it is COCA's *Correctness*. *Attractiveness* of user documentation and its appearance are synonyms. Thus, all those characteristics are covered by COCA's characteristics. What is left outside is *effectiveness* (i.e., the capacity of producing a desired result), and a requirement for software to be *specified* and *documented*. All those three characteristics have no meaning when translated into quality of user documentation perceived from the point of view of the end user.

The last set of quality characteristics is Steidl's quality model for comments in code (Steidl et al. 2013). Steidl's *coherence* (*how comment and code relate to each other*) maps onto COCA's *Correctness* (how user documentation and code relate to each other). Steidl's *completeness* and COCA's *Completeness* are also very similar as they refer to the completeness of information they convey. The remaining two Steidl's characteristics are *usefulness* (the degree of *contributing to system understanding*) and *consistency* (is the language of the comments the same, are the file headers structured the same way, etc.). When translating them into the needs of user documentation readers, they map onto COCA's *Operability* (if user documentation did not contribute to understanding how to use the software, or the language of each chapter was different, *Operability* of such documentation would be low). Thus, the COCA model is also complete from the point of view of Steidl's characteristics. □

4 Review-based evaluation of user documentation

One of the aspects concerning software development is to decide whether a product is ready for delivery or not. A typical activity performed here is acceptance testing. However, this issue concerns not only software, but also user documentation. A counterpart of acceptance testing, when talking about user documentation, is quality evaluation of documentation for the purpose of acceptance. That assessment can be performed taking into account the COCA characteristics and is described below. Another application of the COCA quality model is selection. This kind of evaluation is used to compare two user manuals concerning the same system. The comparison can be performed for a number of purposes, e.g., to decide which method of creation is better (manual writing vs. computer aided) or to select a writer who provides a more understandable description for an audience.

4.1 Goal-Question-Metric approach to evaluation of user documentation

Quality evaluation is a kind of measurement. A widely accepted approach to defining a measurement is Goal-Question-Metric (Solingen and Berghout 1999) (GQM for short). It will be used here to describe quality evaluation when using the COCA quality model.

Goal

The measurement goal of quality evaluation of user documentation can be defined in the following way:

Analyze the user documentation for the purpose of its acceptance with respect to Completeness, Operability, Correctness, and Appearance, from the point of view of the end-user in the context of a given software system.

Table 4 Questions assigned to the COCA characteristics

Question
<p><i>Completeness</i></p> <p>To what extent does the user documentation covers all the functionality provided by the system with the needed level of detail?</p> <p>To what extent does the user documentation provides information which is helpful in deciding whether the system is appropriate for the needs of prospective users?</p> <p>To what extent does the user documentation contains information about how to use it with effectiveness and efficiency?</p>
<p><i>Operability</i></p> <p>To what extent is the user documentation easy to use and helpful when operating the system documented by it?</p>
<p><i>Correctness</i></p> <p>To what extent does the user documentation provides correct descriptions with the needed degree of precision?</p>
<p><i>Appearance</i></p> <p>To what extent is the information contained in the user documentation presented in an aesthetic way?</p>

UC: Evaluation of user documentation

Main scenario:

1. Review Leader creates, on behalf of Decision Maker, an *Evaluation Mandate*. He also prepares *Evaluation Forms*.
2. Experts assess the *user documentation* from the point of view of the quality characteristics assigned to them (e.g. Completeness and Correctness) and fill in the *Evaluation Forms*.
3. Review leader gets the *Evaluation Forms*.
4. Prospective Users assess the *user documentation* from the point of view of the quality characteristics assigned to them (e.g. Operability and Appearance) and fill in the *Evaluation Forms*.
5. Review Leader collects the *Evaluation Forms*, determines the final grade and writes her/his *Evaluation Report*.

Exceptions:

- 3.A. Experts' evaluation is negative.
 - 3.A.1. Go to step 5.

Fig. 1 Procedure for evaluation of user documentation

Questions

Each of the COCA characteristics can be assigned a number of questions which refine the measurement goal. Those questions should cover the quality aspects and documentation themes one is interested in (see justification to Claim 2). Table 4 presents the questions that, from our point of view, are the most important. We hope that they will also prove important in many other settings. Obviously, one can adapt those questions to one's needs.

At first glance, it may appear that the question assigned to Operability is too wide when compared to the definition of Operability (Definition 4), as the definition excludes the completeness and correctness problems. That exclusion is not necessary when the

evaluation procedure first checks Completeness and Correctness, and initiates Operability evaluation only when those checks are successful (see Fig. 1).

Metrics

When evaluating user documentation, two types of quality indicators, also called metrics, can be used: *subjective* and *objective*.

Subjective quality indicators provide information on what people think or feel about the quality of a given documentation. Usually, they are formed as a question with a 5-grade Likert scale. Taking into account the questions in Table 4 (*To what extent...*), the scale could be as follows: *Not at all* (*N* for short), *Weak* (*w*), *Hard to say* (?), *Good enough* (*g*), *Very good* (*VG*). The results of polling can be presented as a vector of 5 integers [*#N*, *#w*, *#?*, *#g*, *#VG*], where *#x* denotes the number of responses with answer *x*. For example, vector [0, 1, 2, 3, 4] means that no one gave the answer *Not at all*, 1 participant gave the answer *Weak*, etc. (this resembles the quality spectrum mentioned by Kaiya et al. (2008)). These kinds of vectors can be normalized to the relative form, which presents the results as a percentage of the total number of votes. For example, the mentioned vector can be transformed to the following relative form [0, 10, 20, 30, 40 %]. This form of representation should be accompanied by the total number of votes that would allow one to return to the original vector.

Objective quality indicators are usually the result of an evaluation experiment and they strongly depend on the design of the experiment. For instance, one could evaluate the Operability of user documentation by preparing a test for subjects participating in the evaluation, asking the subjects to take an open-book examination (i.e., having access to the documentation), and measuring the percentage of correct answers or time used by the subjects.

4.1.1 Interpretation

The fourth element of GQM is interpretation of measurement results. Interpretation requires reference data, against which the obtained measurement data can be compared. Reference data represent a population of similar objects (in our case, user manuals), and they are called a quality profile. In the case of *subjective quality indicators* both the profile and measurement data should be represented in the relative form—this allows one to compare user manuals evaluated by different numbers of people. An example of a quality profile for user manuals is presented in Table 6.

4.2 Evaluation procedure

The proposed evaluation procedure is based on *Management Reviews* of IEEE Std. 1028:2008. This type of review was selected on the grounds that it is very general and can be easily adapted to any particular context.

Moreover, the proposed procedure applies very well to quality management activities undertaken within the framework of PRINCE2 (OGC 2009). PRINCE2 is a project management methodology developed under the auspices of UK's Office of Government Commerce (OGC). Quality management is the central theme of PRINCE2. It is based on two pillars: *Product Description* and *Quality Register*. *Product Description* (one for each product being a part of project output) specifies not only the product's purpose and its composition, but also the quality criteria (with their tolerances), quality methods to be used, and the roles to be played when using the quality methods. In PRINCE2, quality methods are split into two categories:

- in-process methods: they are the means by which quality can be *built into* the products—these are out of scope of this paper,
- appraisal methods: using them allows the quality of the finished products to be assessed—these are what the proposed evaluation procedure is concerned with.

Quality Register is a place (database) where the records concerning planned or performed quality activities are stored.

4.2.1 Roles

The following roles participate in user documentation evaluation:

- *Decision Maker* uses results from the evaluation to decide whether user documentation is appropriate for its purpose or not.
- *Prospective User* is going to use the system documented by the user documentation. For evaluation purposes, it is important that a *Prospective user* does not yet know the system. This lack of knowledge about the system is, from the evaluation point of view, an important attribute of a person in this role.
- *Expert* knows the system very well, or at least its requirements if the system is not ready yet.
- *Review Leader* is responsible for organizing the evaluation and preparing a report for the *Decision Maker*.

4.2.2 Input

The following items should be provided before examining the user documentation:

1. Evaluation mandate for *Review Leader* (see below)
2. Evaluation forms for *Prospective Users*, *Experts* and *Review Leader* (Appendix 2 contains an example of such a form)
3. User documentation under examination
4. Template for an evaluation report (see Appendix 3)

Evaluation Mandate is composed of five parts (an example is given in Appendix 1):

- *Header* besides auxiliary data such as id, software name, file name, etc., it includes the purpose, scope and the evaluation approach:
 - *Purpose of examination* There are two variants: *Acceptance* and *Selection*.
 - *Scope of evaluation* The evaluation can be based on *exhaustive reading* (one is asked to read the whole document) or *sample reading* (reading is limited to a selected subset of chapters). *Sample reading* allows saving effort but makes evaluation less accurate.
 - *Evaluation approach* Depending on available time and resources, different approaches to evaluation can be employed. One can decide to organize a physical meeting or use electronic communication only. Furthermore, the examination can be carried out individually or in groups (e.g., Wideband Delphi (McConnell 2006)). Each meeting can be supported by a number of forms (e.g., *evaluation forms*) and guidelines which should be available before the examination.
- *Evaluation grades* These grades depend on the purpose of the examination. In the case of *Acceptance* evaluation, typical grades are the following: *accept*, *accept with minor*

revision (necessary modifications are very easy to introduce and no other evaluation meeting is necessary), *accept with major revision* (identified defects are not easy to fix and a new version should go through another evaluation), *reject* (quality of the submitted documentation is unacceptable and other corrective actions concerning the staff or process of writing must be taken). These grades can be given on the basis of evaluation data presented together with the population profile. In the case of *Selection* between variants *A* and *B* of the documentation, the grades can be based on the 5-grade scale: *variant A when compared to variant B is definitely better/rather better/hard to say/rather worse/definitely worse*.

- *Selection of quality questions* One should choose quality questions (see Table 4) to be used during evaluation. Each question should be assigned to roles taking into account the knowledge, experience and motivation of people assigned to each role. For example, it is hard to expect from people who do not know the system (or requirements) that they decide whether *user documentation* describes all the functionality supported by the system; thus, evaluation of *Completeness* in such conditions may provide insignificant results.

Evaluation Mandate can be derived from information available in project documentation. For example, a project in which PRINCE2 (OGC 2009) is used should contain a *Product Description* for user documentation. An *Evaluation Mandate* can be derived from that description. In PRINCE2 *Product Description* contains, among others, *Quality Criteria* and *Quality Method* (see Appendix A.17 in OGC (2009)). The *Scope of evaluation* and *Evaluation approach* can be derived from *Quality Method*, and *Selection of quality questions* follows from *Quality Criteria*. *Purpose of examination* usually will be set to *Acceptance* (*Selection* will be used only in research-like projects when one wants to compare different methods or tools).

4.2.3 Evaluation

Activities required to evaluate user documentation are presented in Fig. 1 in the form of a use case (Cockburn 2000). Use cases seem to be a good option as they can be easily understood, even by IT-laymen.

4.2.4 Quality evaluation procedure versus management reviews

The proposed procedure differs from the classical *Management review* (IEEE 2008) in the following aspects:

- The proposed procedure has a clear interface to PRINCE2's *Product Description* through *Evaluation Mandate* (see Sect. 4.2.2).
- Experts (their counterparts in *Management Review* are called Technical staff) and Prospective Users (in *Management Review* they are called User representatives) have clearly defined responsibilities (see Fig. 1).
- Decision making is based on clearly described multiple criteria accompanied by a quality profile describing previously evaluated documents (see Interpretation of Sect. 4.1 and Appendix 3).

4.3 Quality profile for user documentation

In the case of *Acceptance*, it is proposed that a given user documentation is compared with other user manuals created by a given organization (e.g., company) or available on the

market. Instead of comparing user documentation at hand with n other documents, one by one, it is proposed that those n documents are evaluated, a quality profile describing an average user documentation is created and the given user documentation is compared with the quality profile (see Table 6).

To give an example, a small research has been conducted, the goal of which can be described as follows:

Analyze a set of user manuals for the purpose of creating a quality profile from the point of view of end-users and in the context in which the role of end-users is played by students and the role of Experts is played by researchers and Ph.D. students.

The evaluation experiment was designed in the following way:

- For each considered user manual, one of the authors played the role of Review Leader, three Experts were assigned from Ph.D. students and staff members, and 16–17 students were engaged to play the role of Prospective Users.
- The evaluation was performed as a controlled experiment based on the procedure described in Fig. 1.
- The evaluation time available to Prospective Users was limited to 90 min. None of the subjects exceeded the allotted time.
- The evaluated user manuals were selected to describe commercial systems and concerned a domain which was not difficult to understand for the subjects playing the role of Prospective Users. The user manuals were connected with the products available on the Polish market which are presented in Table 5. For *Plagiarism.pl*, *nSzkola*, and *Hermes* the whole user manual was evaluated; in all the other cases, only selected chapters describing a consistent subset of functionality went through review.

The resulting quality profile is presented in Table 6 and the data collected during evaluation are available in Appendix 4. As the role of experts was played by Ph.D. students and staff members, who knew only some of the systems used in the experiment, the percentage of g (good) and VG (very good) grades shown in Table 6 (questions Q1 and Q5) should be regarded rather as upper limits (real experts could identify some functionality provided by the system which was not covered in the evaluated users manuals, or some additional incorrect descriptions).

How to use the data of a quality profile such as the one presented in Table 6 is another question. When making a final decision (to accept or reject a user manual) one can use one of many multi-attribute decision making methods and tools (there are many of them—see e.g., Zanakis et al. 1998; Figueira et al. 2005). For instance one could use the notion of dominance and require that a given user manual gets a score, for every criterion (characteristic), not worse than a given threshold. Such a threshold could be calculated, for instance, as a percentage of g and VG answers to each question. It is also possible to infer thresholds from a historical database, providing that the database contains both evaluation answers and final decisions (or customer opinions).

When using the profile presented in Table 6, one should be aware that all the evaluated documents are connected with educational software (see Table 5). So, one must be careful when using the presented profile in other contexts. We believe that a profile, such as of Table 6 can be useful especially when a company or a project does not have its own quality profile. To support this we established a web page with results from ongoing evaluations ².

² <http://coca.cs.put.poznan.pl/>

Table 5 List of evaluated user manuals (pages are counted without cover page and table of contents; last column presents number of Experts and Users participating in an evaluation)

User documentation	Description	Pages	Experts/Users
<i>Plagiarism.pl—Manual for individual user</i> (in Polish <i>Plagiat.pl—Instrukcja użytkownika indywidualnego</i>)	The system allows detection of plagiarism in different types of documents, e.g., M.Sc. thesis	13	3/16
<i>Getting Started with Deanery.XP 9.65.5.0</i> (in Polish <i>Podstawy obsługi Dziekanatu.XP</i>)	Supports staff of a dean office in management of students at a university	19	3/17
<i>Optivum Sekretariat—User manual</i> (in Polish <i>Sekretariat Optivum—Podręcznik użytkownika programu</i>)	Supports management of a primary and secondary school	25	3/17
<i>User manual for nSzkoła platform—Student's panel</i> (in Polish <i>Instrukcja obsługi Platformy nSzkoła—Panel Ucznia</i>)	Allows students to read records in an electronic log	16	3/16
<i>Sekretariat DDJ 6.8</i> (in Polish <i>Sekretariat DDJ 6.8</i>)	Supports management of a school	21	3/16
<i>LangSystem 4.2.5—User documentation</i> (in Polish <i>LangSystem 4.2.5—Dokumentacja użytkownika</i>)	Supports management of a school of foreign languages	22	3/17
<i>SchoolManager—User manual</i> (in Polish <i>School Manager—Podręcznik użytkownika</i>)	Supports management of a school of foreign languages	27	3/17
<i>User manual for Hermes 2012</i> (in Polish <i>Instrukcja obsługi aplikacji HERMES 2012</i>)	Collecting data about examinations concerning professional qualifications	21	3/16
<i>E-grades: Electronic log</i> (in Polish <i>Dziennik elektroniczny e-oceny</i>)	Allows students to read records in an electronic log	23	3/16

Table 6 An exemplary quality profile (9 user manuals, 3 experts, 16–17 prospective users per manual *N* not at all, *w* weak, *?* hard to say, *g* good enough *VG* very good)

Id	Questions	N	w	?	g	VG
Completeness						
					<i>responsible: Expert</i>	
Q1	To what extent does the user documentation cover all the functionality provided by the system with the needed level of detail?	3.7%	18.5%	29.6%	44.4%	3.7%
Q2	To what extent does the user documentation provide information which is helpful in deciding whether the system is appropriate for the needs of prospective users?	0.0%	3.7%	11.1%	55.6%	29.6%
					<i>responsible: Prospective User</i>	
Q3	To what extent does the user documentation contain information about how to use it with effectiveness and efficiency?	6.1%	9.5%	7.4%	50.0%	27.0%
					<i>responsible: Prospective User</i>	
Operability						
Q4	To what extent is the user documentation easy to use and helpful when operating the system documented by it?	1.4%	6.8%	14.9%	48.0%	29.1%
					<i>responsible: Expert</i>	
Correctness						
Q5	To what extent does the user documentation provide correct descriptions with the needed degree of precision?	0.0%	18.5%	25.9%	44.4%	11.1%
					<i>responsible: Prospective User</i>	
Appearance						
Q6	To what extent is the information contained in the user documentation presented in an aesthetic way?	1.4%	12.2%	12.2%	49.3%	25.0%

UC: Documentation Evaluation Test*Main scenario:*

1. Experts individually read *user documentation*, create *Questions* and pass them to Review Leader.
2. Review Leader cleans the *Questions* submitted by the Experts (i.e. removes duplicates, corrects spelling, etc.).
3. Review Leader prepares a *Knowledge Test* by random selection of *Questions*.
4. Prospective Users, to assess *Operability*, take an open-book *Knowledge Test* (the book is the *user documentation*).
5. Review Leader writes a *Review Report* concerning the *user documentation*.

Extensions:

- 3.A. Review Leader realizes that the number of *Questions* is too small.
 - 3.A.1. Review Leader asks one more Expert to perform step 1.

Fig. 2 The DET procedure**5 Empirical evaluation of operability**

To evaluate a user manual experimentally, one can use a form of browser evaluation test (BET) (Wellner et al. 2005). The BET method was developed to evaluate the quality of meeting browsers based on a video recording of a meeting. In such an evaluation each subject is given a list of complementary assertions (one is true and the other is false), and must identify which of the two is true (e.g., one is *Susan says the footstool is not expensive* and the other is *Susan says the footstool is expensive* (Wellner et al. 2005)). Obviously, by making lucky guesses one can get a score of about 50 %. From our point of view this is unacceptable. To help this, a variant of BET was developed (see below) which is oriented

toward evaluation of user documentation (it is called documentation evaluation test—DET) and by guessing one can get a score of about 25 %. The DET procedure is presented in Fig. 2.

5.1 DET questions

Questions are very important for the effectiveness of the DET procedure. An exemplary question is presented in Table 7. A DET *question* consists of a *theme* (e.g., *The following items are included into a similarity report*) and four proposed answers of which one is correct and the other three are false. Every question is accompanied by an auxiliary statement (*I could not find the answer*) which is to be evaluated by the subject (true/false). That statement allows subjects to say that for some reasons they failed when trying to find the answer. Questions with answers and additional statements are used to create a *Knowledge Test* which is presented to subjects during an evaluation.

Table 7 Exemplary question

<i>Question no 2</i>	
The following items are included into a similarity report:	
<i>Choose one of the proposed answers:</i>	<i>Correct?</i>
A) Info about whether a given document is plagiarised	
B) Similarity coefficients and a list of similar documents	
C) Similarity coefficients, a list of similar documents and whether a given document is plagiarised	
D) Similarity coefficients, a list of similar documents and fragments of the document which have been found in another document	
<i>The answer is in the user documentation on page:</i>	
<i>I could not find the answer:</i>	

When analyzing questions provided by Experts at early stages of this research, we identified a number of weaknesses, which are unacceptable:

- W1. Some choices were synonyms, e.g., *month* and *1/12 of year*.
- W2. Some choices were answers to other questions.
- W3. Some questions were suggesting a number of choices (e.g., *The following values are correct ISBN numbers*).
- W4. Some references to the user interface were imprecise, especially when elements with the same name occur multiple times in a different context.
- W5. Some choices did not require the user manual to make a selection—it was enough to use general knowledge.

To cope with these weaknesses, a set of guidelines was formulated. Here they are:

- the choices of questions should not contain a synonym of any other choice (addresses weakness W1).
- the choices of questions should not contain an answer to any other question (addresses weakness W2).
- questions should not suggest a number of choices (addresses weakness W3).
- references to the user interface must be unambiguous (addresses weakness W4).
- selecting a choice must require information contained in the user documentation (addresses weakness W5).

Table 8 Results of DET evaluation

User documentation	No. of participants	No. of pages	Average answer time (min)	No. of questions	Average percentage of correct answers (%)
Plagiarism.pl	16	13	39	29	82.97
Deanery.XP	17	19	40	28	86.97
Optivum Secretariat	17	25	61	30	76.47
LangSystem	17	22	52	30	81.76
Hermes	16	21	52	28	77.01
Total	83	100	244	145	

Table 9 Preparation of questions for DET evaluation

User documentation	No. of experts	Final/total no. of questions	Total time of writing questions (min)	Average time for one final question (min)
Plagiarism.pl	3	29/31	329	11.3
Deanery.XP	3	28/31	365	13.0
Optivum Secretariat	3	30/32	264	8.8
LangSystem	3	30/31	350	11.7
Hermes	3	28/30	224	7.7
Average		29/31	306.4	10.6

5.2 Case studies

To characterize the DET method, we have analyzed five user manuals with the aim of presenting an example of how such an evaluation could be conducted. Each user manual was assessed with the following purpose in mind:

Analyze the user manual for the purpose of quality evaluation with respect to Operability, from the point of view of end-users in the context of Ph.D. students playing the role of Experts and students as Prospective Users.

The evaluation experiment was designed in similarly to the one presented in Sect. 4.3. The evaluation procedure used in the experiment is described in Fig. 2 and the manuals are listed in Table 8. All of them had been checked earlier for Completeness and Correctness by Experts (that role was played by three researchers and Ph.D. students) and it was executed as a one-person review (see Appendix 4 for results of the Completeness and Correctness checks).

The data collected during the evaluation are summarized in Table 8. The average speed of reading a manual by a Prospective User was about 4 pages per 10 min and the average percentage of correct answers was about 81 %. Table 9 contains data concerning preparation of questions. There are two numbers referring to questions: total number of questions and final number of questions. The first one describes total number of questions proposed by the experts. Some of those questions overlapped, so the final number of questions included in the *Knowledge test* was a bit smaller (e.g., for *Plagiarism.pl* 31 questions have been proposed and 29 of them have been included into the *Knowledge test*). The average speed of writing questions is about 6 questions per hour. One can use those data as reference values when organizing one's own DET evaluation.

6 Related work

One could consider the 265nm series of ISO/IEC standards (ISO/IEC 2008, 2009; ISO/IEC/IEEE 2011, 2012a, b) as a quality model for user documentation as those standards present a number of aspects concerning the quality of user documentation. Unfortunately, those aspects do not constitute an orthogonal quality model. For example, *completeness of information* contains *error messages* as its sub-characteristic. On the other hand, *safety* is described as containing *warnings and cautions*. Thus, the scope of *completeness of information* overlaps the scope of *safety*. Another example is *Technical accuracy*, which is described as *consistency with the product*, and *Navigation and display* which requires that *all images or icons [...] are correctly mapped to the application*—those two characteristics overlap. A similar relation exists between *Technical accuracy* and *Accuracy of information*, which—according to its description—should *accurately reflect the functions* of the software. Thus, the intention of the authors of the standards was not to present an orthogonal quality model, but rather the way in which user documentation should be assessed.

Markel (2012) presented *eight measures of excellence* which are important in technical communication: *honesty, clarity, accuracy, comprehensiveness, accessibility, conciseness, professional appearance* and *correctness*. Each item on the list is described, and why it is important from the quality perspective is explained. Unfortunately, there is no information on how to evaluate the presented *measures*. Moreover, some of these *measures* overlap, i.e., both *honesty* and *accuracy* emphasize the importance of not misleading the readers. Moreover, *honesty* is not a characteristic of a user manual but rather a relation between a writer and his/her work (a reviewer can only observe inconsistency between a user manual and the corresponding software but is not able to say if those defects follow from bad will or whether they occurred by chance).

Allwood and Kalén (1997) described the process of assessing the usability of a user manual by reading it and noting difficulties. During the evaluation, participants are asked to rate, for each page of a user manual, its *usability, comprehensibility, readability*, and how *interesting* and *stimulating* it is. Again, the orthogonality of the proposed model is questionable as *usability* strongly depends on the *comprehensibility* of user documentation. Moreover, if the proposed model is to be complete, *usability* should cover operability. As operability depends on *readability* (if a user document is not readable, then it will take longer to get information from it, and thus, its operability will suffer), *usability* and *readability* overlap.

Other quality models considered in this paper are Ortega's systemic quality model and Steidl's characteristics for code comments. They do not directly relate to user documentation but contain quality characteristics that can be “translated” to the context of user documentation. We used them to examine completeness of the COCA model (see Sect. 3, justification for Claim 2).

7 Conclusions

This paper presents the COCA quality model, which can be used to assess the quality of user documentation. It consists of only four characteristics: Completeness, Operability, Correctness, and Appearance. The model is claimed to be orthogonal and complete, and justification for the claims are presented in Sect. 3. As quality evaluation resembles measurement, the GQM approach (Solingen and Berghout 1999) was used to define the goal of evaluation, the questions about quality one should be interested in, and the quality indicators which, when compared to the quality profile for a given area of application, help

to answer those questions. The empirical data (quality profile) have been obtained by evaluating nine user manuals available on the Polish market, which concern education-oriented software (see Table 6). The collected data are interesting. Although the evaluated user manuals concern commercial software, their quality is not very high. For instance, only in 48.1 % of the cases, the Experts evaluated the manuals as *good* or *very good* with respect to functional completeness of the examined user documentation (question Q1 in Table 6); in 22.2 % of the cases, the answer was *weak* or *not-at-all*.

Quality of user documentation can be evaluated with the COCA model using two approaches: *pure review* based on *Management Review* of IEEE Std. 1028:2008 (see Sect. 4.2), or *mixed evaluation* where Completeness, Correctness, and Appearance are evaluated using *Management Review*, and Operability is evaluated experimentally using the DET method proposed in Sect. 5. That method is based on questions prepared by experts. The operability indicator is defined as the percentage of correct answers given by a sample of prospective users. Empirical data concerning DET-based evaluation show that, on average, there are about 1.5 questions per page of user documentation (see Table 8), and on average, it takes an expert about 10 mins to prepare one question. In the DET-based evaluation, prospective users read a user manual at the average speed of about 25 pages per hour, and for documentation concerning commercially available software, the average percentage of correct answers is between 77 and 87%.

Future work should mainly focus on further development of the quality profile, of which an initial version is presented in Sect. 4.3 (Table 6) and Sect. 5.2 (the rightmost column of Table 8). It would also be interesting to investigate Operability indicators based on readability formulae such as SMOG (McLaughlin 1969) or the Fog Index (Gunning 1952) (the Fog Index was used by Khamis to assess the quality of source code comments (Khamis et al. 2010); a similar approach could be applied to user manuals).

Acknowledgments We would like to thank Magdalena Deckert, Sylwia Kopczyńska, Jakub Jurkiewicz, Michał Maćkowiak, Mirosław Ochodek, Konrad Siek and Wojciech Wojciechowicz for their help in performing the experiments. We are also thankful to the anonymous reviewers for helpful comments that allowed us to improve the paper. This work has been supported by the Polish National Science Centre based on the decisions DEC-2011/03/N/ST6/03016.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Appendix 1: Evaluation mandate—an example

ID	EM20130610
Software	ROPS: Registration and evaluation of curricula
Documentation name	User Manual
Documentation version	20130206
Filename	ROPS-UserManual-20130206.pdf
Evaluation deadline	12 June 2013
Purpose	Acceptance or rejection of the user documentation
Scope	Whole document
Evaluation approach	Individual review + evaluation form EF20130610

Evaluation grades

Final grades:

- *accept*
- *accept with minor revision*—necessary modifications are very easy to introduce and no other evaluation meeting is necessary
- *accept with major revision*—identified defects are not easy to fix and a new version should go through another evaluation procedure
- *reject*—quality of the submitted documentation is unacceptable and other corrective actions concerning the staff or process of writing must be taken

Standard answers to questions (5-level Lickert):

- *Not at all (N for short)*
- *Weak (w)*
- *Hard to say (?)*
- *Good enough (g)*
- *Very good (VG)*

Selection of quality questions

Question	Expert	Prosp. user
Completeness		
To what extent does the user documentation cover all the functionality provided by the system with the needed level of detail?	✓	
To what extent does the user documentation provide information which is helpful in deciding whether the system is appropriate for the needs of prospective users?	✓	
To what extent does the user documentation contain information about how to use it with effectiveness and efficiency?		✓
Operability		
To what extent is the user documentation easy to use and helpful when operating the system documented by it?		✓
Correctness		
To what extent does the user documentation provide correct descriptions with the needed degree of precision?	✓	
Appearance		
To what extent is the information contained in the user documentation presented in an aesthetic way?		✓

Appendix 2: Evaluation form for prospective users—an example

ID	EM20130610
Software	ROPS: Registration and evaluation of curricula
Documentation name	User Manual
Documentation version	20130206
Filename	ROPS-UserManual-20130206.pdf
Evaluation deadline	12 June 2013
Name and surname	Eva Smith

Question	N	w	?	g	VG
<i>Completeness</i>					
To what extent does the user documentation contain information about how to use it with effectiveness and efficiency?					
<i>Operability</i>					
To what extent is the user documentation easy to use and helpful when operating the system documented by it?					
<i>Appearance</i>					
To what extent is the information contained in the user documentation presented in an aesthetic way?					

Comments and remarks:

Id	Place	Char.	Description	Type	Priority
----	-------	-------	-------------	------	----------

Appendix 3: Evaluation report—an example

ID	EA20130611
Software	ROPS: Registration and evaluation of curricula
Documentation name	User Manual
Documentation version	20130206
Filename	ROPS-UserManual-20130206.pdf
Evaluation deadline	12 June 2013
Evaluation date	11 June 2013
Purpose	Acceptance of rejection of the user documentation
Scope	Whole document
Evaluation approach	Individual review + evaluation form EF20130610

Results

Final grade		reject				
Question		N	w	?	g	VG
Completeness		<i>responsible: Expert (1)</i>				
To what extent does the user documentation cover all the functionality provided by the system with the needed level of detail?	ROPS	0.0%	0.0%	0.0%	0.0%	100.0%
	Profile	3.7%	18.5%	29.6%	44.4%	3.7%
To what extent does the user documentation provide information which is helpful in deciding whether the system is appropriate for the needs of prospective users?	ROPS	0.0%	0.0%	0.0%	100.0%	0.0%
	Profile	0.0%	3.7%	11.1%	55.6%	29.6%
		<i>responsible: Prospective User (3)</i>				
To what extent does the user documentation contain information about how to use it with effectiveness and efficiency?	ROPS	0.0%	0.0%	33.3%	66.7%	0.0%
	Profile	6.1%	9.5%	7.4%	50.0%	27.0%
Operability		<i>responsible: Prospective User (3)</i>				
To what extent is the user documentation easy to use and helpful when operating the system documented by it?	ROPS	33.3%	33.3%	33.3%	0.0%	0.0%
	Profile	1.4%	6.8%	14.9%	48.0%	29.1%
Correctness		<i>responsible: Expert (1)</i>				
To what extent does the user documentation provide correct descriptions with the needed degree of precision?	ROPS	0.0%	0.0%	0.0%	100.0%	0.0%
	Profile	0.0%	18.5%	25.9%	44.4%	11.1%
Appearance		<i>responsible: Prospective User (3)</i>				
To what extent is the information contained in the user documentation presented in an aesthetic way?	ROPS	0.0%	0.0%	33.3%	66.7%	0.0%
	Profile	1.4%	12.2%	12.2%	49.3%	25.0%

Comments and remarks

Id	Place	Char.	Description	Author	Type	Priority
1	p. 1	Coml.	Data about user documentation (name, version, etc.) are missing.	E1	missing	major
2	all	Coml.	No page number.	E1	missing	major
3	p. 3	Coml.	Role Guest is not described.	E1	missing	major
4	p. 3	Coml.	Abbr. OEK and KRK are not explained.	E1, P2	missing	major

Evaluation team

Decision Maker *Jerzy Nawrocki*
 Review Leader *Bartosz Alchimowicz*
 Experts (1) *E1 - John Smith*
 Prospective Users (3) *P1 - Eva Smith, P2 - Adam Smith, P3 - Peter Smith*

Appendix 4: Evaluation report for profile

Purpose	data collection
Evaluation approach	individual review + evaluation form

Results

	Software	N	w	?	g	VG
Completeness						
<i>responsible: Expert</i>						
To what extent does the user documentation cover all the functionality provided by the system with the needed level of detail?	Plagiarism.pl	0	0	1	1	1
	Deanery.XP	0	0	1	2	0
	Optivum Secr.	0	1	2	0	0
	nSzkoka	0	2	0	1	0
	Secr. DDJ	0	0	1	2	0
	LangSystem	1	1	1	0	0
	SchoolMgr.	0	0	1	2	0
	Hermes	0	1	1	1	0
	E-oceny	0	0	0	3	0
To what extent does the user documentation provide information which is helpful in deciding whether the system is appropriate for the needs of prospective users?	Plagiarism.pl	0	0	0	1	2
	Deanery.XP	0	0	1	2	0
	Optivum Secr.	0	0	1	2	0
	nSzkoka	0	0	1	2	0
	Secr. DDJ	0	0	0	2	1
	LangSystem	0	1	0	1	1
	SchoolMgr.	0	0	0	1	2
	Hermes	0	0	0	3	0
	E-oceny	0	0	0	1	2
<i>responsible: Prospective User</i>						
To what extent does the user documentation contain information about how to use it with effectiveness and efficiency?	Plagiarism.pl	3	1	1	11	0
	Deanery.XP	3	3	1	8	2
	Optivum Secr.	2	4	3	8	0
	nSzkoka	0	1	1	11	3
	Secr. DDJ	0	1	2	6	7
	LangSystem	1	2	1	7	6
	SchoolMgr.	0	2	2	8	5
	Hermes	0	0	0	8	8
	E-oceny	0	0	0	7	9
Operability						
<i>responsible: Prospective User</i>						
To what extent is the user documentation easy to use and helpful when operating the system documented by it?	Plagiarism.pl	0	5	2	7	2
	Deanery.XP	0	0	2	9	6
	Optivum Secr.	2	2	2	6	5
	nSzkoka	0	1	2	10	3
	Secr. DDJ	0	1	4	8	3
	LangSystem	0	0	2	12	3
	SchoolMgr.	0	0	3	7	7
	Hermes	0	0	3	7	6
	E-oceny	0	1	2	5	8
Correctness						
<i>responsible: Expert</i>						
To what extent does the user documentation provide correct descriptions with the needed degree of precision?	Plagiarism.pl	0	0	1	1	1
	Deanery.XP	0	0	0	2	1
	Optivum Secr.	0	2	1	0	0
	nSzkoka	0	1	0	2	0
	Secr. DDJ	0	0	0	2	1
	LangSystem	0	1	2	0	0
	SchoolMgr.	0	0	1	2	0
	Hermes	0	0	1	2	0
	E-oceny	0	1	1	1	0
Appearance						
<i>responsible: Prospective User</i>						
To what extent is the information contained in the user documentation presented in an aesthetic way?	Plagiarism.pl	2	3	2	9	0
	Deanery.XP	0	4	2	10	1
	Optivum Secr.	0	4	1	10	2
	nSzkoka	0	1	0	8	7
	Secr. DDJ	0	2	3	6	5
	LangSystem	0	3	3	6	5
	SchoolMgr.	0	0	4	11	2
	Hermes	0	1	3	5	7
	E-oceny	0	0	0	8	8

Evaluation team

System	Review Leader	Experts	Prospective Users
Plagiarism.pl	1	3	16
Deanery.XP	1	3	17
Optivum Secretariat	1	3	17
nSzkoo	1	3	16
Secretariat DDJ	1	3	16
LangSystem	1	3	17
SchoolManager	1	3	17
Hermes	1	3	16
E-oceny	1	3	16

References

- Allwood, C. M., & Kalén, T. (1997). Evaluating and improving the usability of a user manual. *Behaviour & Information Technology*, 16(1), 43–57.
- Cockburn, A. (2000). *Writing effective use cases* (1st ed.). Boston, MA: Addison-Wesley Longman Publishing Co. Inc.
- Figueira, J., Greco, S., & Ehr Gott, M. (2005). *Multiple criteria decision analysis: State of the art surveys*, Vol. 78. Berlin: Springer.
- Fisher, J. (2001). User satisfaction and system success: considering the development team. *Australasian Journal of Information Systems*, 9(1).
- Gunning, R. (1952). *The technique of clear writing*. New York: McGraw-Hill International.
- Hazlett, R. (2003). Measurement of user frustration: A biologic approach. In *CHI '03 extended abstracts on human factors in computing systems*, CHI EA '03 (pp. 734–735). New York, NY: ACM. doi:10.1145/765891.765958.
- Huang, S., & Tilley, S. (2003). Towards a documentation maturity model. In *Proceedings of the 21st annual international conference on documentation* (pp. 93–99). ACM.
- IEEE. (2008). IEEE Std 1028–2008, IEEE Standard for Software Reviews and Audits. Institute of Electrical and Electronics Engineers.
- ISO/IEC. (2001). *ISO/IEC 9126–1:2001—Software engineering—Product quality—Part 1: Quality model*. Geneva: International Organization for Standardization.
- ISO/IEC. (2005). *ISO/IEC 25000:2005—Software engineering—Software product quality requirements and evaluation (SQuaRE)—Guide to SQuaRE*. Geneva: International Organization for Standardization.
- ISO/IEC. (2008). *ISO/IEC 26514:2008—Systems and software engineering—Requirements for designers and developers of user documentation*. Geneva: International Organization for Standardization.
- ISO/IEC. (2009). *ISO/IEC 26513:2009—Systems and software engineering—Requirements for testers and reviewers of user documentation*. Geneva: International Organization for Standardization.
- ISO/IEC (2011). *ISO/IEC 25010:2011—Systems and software engineering—Systems and software quality requirements and evaluation (SQuaRE)—System and software quality models*. Geneva: International Organization for Standardization.
- ISO/IEC/IEEE. (2010). *ISO/IEC/IEEE 24765:2010—Systems and software engineering—Vocabulary*. Geneva: International Organization for Standardization.
- ISO/IEC/IEEE. (2011). *ISO/IEC 26512:2011—Systems and software engineering—Requirements for acquirers and suppliers of user documentation*. Geneva: International Organization for Standardization.
- ISO/IEC/IEEE. (2012a). *ISO/IEC 26511:2012—Systems and software engineering—Requirements for managers of user documentation*. Geneva: International Organization for Standardization.
- ISO/IEC/IEEE. (2012b). *ISO/IEC 26515:2012—Systems and software engineering—Developing user documentation in an agile environment*. Geneva: International Organization for Standardization.
- Jung, H. W., Kim, S. G., & Chung, C. S. (2004). Measuring software product quality: A survey of iso/iec 9126. *IEEE Software*, 21(5), 88–92.
- Kaiya, H., Sato, T., Osada, A., Kitazawa, N., & Kaijiri, K. (2008). Toward quality requirements analysis based on domain specific quality spectrum. In *Proceedings of the 2008 ACM symposium on applied computing SAC '08* (pp. 596–601). New York, NY: ACM.
- Khamis, N., Witte, R., Rilling, J. (2010). Automatic quality assessment of source code comments: The javadocminer. In *Natural language processing and information systems* (pp. 68–79). Springer.
- Markel, M. (2012). *Technical communication*. Bedford /St. Martin's.

- McConnell, S. (2006). *Software estimation: Demystifying the black art*. Redmond, WA: Microsoft Press.
- McLaughlin, G. H. (1969). Smog grading: A new readability formula. *Journal of Reading*, 12(8), 639–646.
- Novick, D. G., & Ward, K. (2006a). What users say they want in documentation. In S. Huang, R. Pierce, & J. W. S. Jr. (eds.) *SIGDOC* (pp. 84–91). ACM.
- Novick, D. G., & Ward, K. (2006b). Why don't people read the manual? In S. Huang, R. Pierce, J. W. S. Jr. (eds.) *SIGDOC* (pp. 11–18). ACM.
- OGC. (2009). *Managing successful projects with PRINCE2*. HM Government—Best management practice. Stationery Office.
- Ortega, M., Pérez, M., & Rojas, T. (2003). Construction of a systemic quality model for evaluating a software product. *Software Quality Journal*, 11(3), 219–242.
- Pedraz-Delhaes, A., Aljukhadar, M., & Sénécal, S. (2010). The effects of document language quality on consumer perceptions and intentions. *Canadian Journal of Administrative Sciences/Revue Canadienne des Sciences de l'Administration*, 27(4), 363–375.
- van Solingen, R., Berghout, E., & New York. (1999). *The Goal/Question/Metric Method: A practical guide for quality improvement of software development*. New York: McGraw-Hill.
- Spencer, C. J. (1995). A good user's guide means fewer support calls and lower support costs. *Technical Communication*, 42(1), 52–55.
- Steidl, D., Hummel, B., & Juergens, E. (2013). Quality analysis of source code comments. In *2013 IEEE 21st international conference on program comprehension (ICPC)* (pp. 83–92). IEEE.
- Wellner, P., Flynn, M., Tucker, S., & Whittaker, S. (2005). A meeting browser evaluation test. In: *CHI '92: Proceedings of the SIGCHI conference on human factors in computing systems*. New York, NY: ACM Press.
- Zanakis, S. H., Solomon, A., Wishart, N., & Dublsh, S. (1998). Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research*, 107(3), 507–529.



Bartosz Alchimowicz is a Project Manager of a research project supported by Polish National Science Center at the Faculty of Computing, Poznan University of Technology in Poland. His main fields of interest are quality of documentation, natural language processing and requirements engineering.



Jerzy R. Nawrocki received M.S. degree (1980), Ph.D. degree (1984), and Dr.hab. degree (1994)—all in Informatics and all from the Poznan University of Technology (PUT), Poznan, Poland. Currently, he is a dean of the Faculty of Computing at PUT and a Vice-Chair of IFIP Technical Committee 2: Software Theory and Practice.