Software product certification – an alternative approach for a small software house

M. Pivka
School of Business and Economics, University of Maribor, 62000 Maribor, Slovenia

Abstract

A complete implementation of Software quality system harmonised with ISO 9001 is for a small software house a difficult and expensive approach. Some of ISO 9001 requirements are for small software houses without outsourcing impossible to be fulfilled. An alternative approach to software process certification is software product certification. The present paper represents a brief description of software product quality characteristics and testing procedures, as recommended and required by International standards ISO/IEC 9126 and ISO/IEC 12119. Added value to the software package developer, limitations of this approach and some practical experiences on product certification in Slovenia have been analysed.

1 Definition of the problem

For the purpose of this paper a small software house or a smaller developer is defined as a software house having about 10 employees or less, with more than 50 users, producing typical PC sized applications with less than 100 KLOC. Such a small software house is facing the problems such as: how to finance and implement the Quality Management System (QMS); is it worth the efforts and costs; will QMS be helpful in the productivity etc.

It is clear that small software houses can not implement QMS and harmonise it with international standards such as ISO 9001 without substantial outside and inside resources. The analysis of this problem (Pivka[4]) shows that some basic requirements of the quality system are difficult or even impossible to be met. For instance, the principle of independent control, verification and validation procedures.

Another possibility to meet the universal goal, to evolve quality products, is to test the software product quality. Testing the product quality does not require the implementation of QMS, quality manuals, quality records etc. It requires only general quality characteristics for a product (any type or a specific type) and general testing procedures. Two standards are devoted to this problem: the German DIN 66285:1990[1] and the international standard ISO/IEC 12119:1994[2] which is based on the German standard.

The problem is the following: From the user's or the developer's aspect, is it reasonable to invest into the QMS certification procedures or into the product certification procedures? What added value will a small developer obtain from QMS certification procedures or from product certification procedures? The problem discussed in this paper is the question, when and how a product certification procedure can be an alternative approach for a small software house to produce better software.

2 Software quality characteristics

The software quality characteristics are defined by the international standard ISO/IEC 9216[3]. Six software quality characteristics defined in this standard are:

- Functionality: does it satisfy the stated needs?
- Reliability: how often does it fail?
- Usability: how easy is it to be used?
- Efficiency: how much resources does it need?
- Maintainability: how easy is it to be repaired?
- Portability: how easy is it to be transported?

The standard is limited to the description of these characteristics and the evaluation process and does not provide any professionally accepted metrics for these characteristics. Nevertheless, those characteristics are recognised and accepted in the software community as basic software quality characteristics.

To use this standard in a certification procedure, the certification body needs to have a clear definition for each one of these characteristics, including the accepted metrics for them. As we have no accepted metrics for any type of software products it is obvious, that this standard is impossible to be used directly in the certification procedure.

3 Quality requirements

Who defines the quality requirements for a software product? The developer? The user? The testing laboratory? Standards? It is not easy to answer this question. The quality requirements are defined by all the above mentioned ones and also by others, such as professional regulations,

legislation etc. The question is, who defines what, or which requirement is defined by whom. It is obvious that the government is responsible to define the quality requirements for some systems, such as tax system. We strongly believe that any type of business is responsible to define their own specific quality requirements. For instance: Nobody but a medicine specialist can define the quality requirements for the software used in medicine. However, there are some general quality requirements, recognised for typical software packages such as accounting systems, office communications, utility programs, technical or scientific functions etc. These quality requirements are defined by the German DIN 66285[1] standard and the international standard ISO/IEC 12119:1994[2]. The international standard is based on the German standard and it incorporates 6 quality characteristics quoted in ISO/IEC 9126:1994.

3.1 Quality requirements by ISO/IEC 12119:1994

The quality requirements refer to the product documentation and to programs and data, if they exist. Individual standard requirements are divided to the mandatory and recommended ones. The desirable requirements are those that can not be tested objectively, such as "understandability", "simplicity of application" and similar. By the opinion of some testing opposes, testing covers only the documentation. It shall be immediately emphasised that testing equally includes all the requirements: from the documentation to the programs and data.

The quality requirements are divided into three groups of requirements (criteria) as follows:

- requirements for the product description
- requirements for user's documentation and
- requirements for programs and data.

It should be emphasised once again that the quality requirements for programs and data are equivalent to the requirements for documentation.

3.1.1 Quality requirements for product description

The product description is a brief presentation thereof and is intended for the potential buyer of the product, to get him acquainted with its basic characteristics, functions, purpose etc. before the purchase. It is the constituent part of the product, yet separated from the other documentation. The description shall be clear, understandable, terminological harmonised with the other documentation, all data have to be controllable. It shall include at least the following data on the product:

1. Identifications and indications such as: product ID, version, work task, required configuration etc. are mandatory.

2. Statements on functionality such as: review of functions, specific boundary values etc. are mandatory.

3. Statements on reliability such as: data saving procedures, recoverability, input and output controls, etc. Informations on data saving procedures are mandatory.

4. Statements on usability such as: user's interfaces, user's knowledge of the technical area and operating systems, adaptation to user's needs etc. Some of these data, e.c. user's interface are mandatory.

5. Statements on efficiency such as: response times, resource requirements etc. These data are not mandatory.

6. Statements on maintainability are not mandatory.

7. Statements on portability are not mandatory.

3.1.2 Quality requirements for documentation

The documentation for the user shall include all the data necessary for the application, installation (if necessary) and maintenance of the software product. It shall be faultless, the data shall be in accordance with each other. The requirements are as follows:

1. Perfectness and absence of faults. The documentation intended for the user shall contain all the data and information necessary for the application of the software product. It shall include the description of all functions, stated in the product description and accessible by the user.

All limit (boundary) values, stated in the product description, shall be quoted once again. If the installation can be done by the user himself, the installation instructions shall be included in the document description. These instructions shall include also the minimal and maximal sizes of installed files.

If the maintenance can be done by the user himself, the program maintenance manual shall be included in the documentation. The same refers also to parametring of functions that can be adapted to the user's requirements.

2. Correctness (exactness). All the information in the documentation shall be correct. The representation thereof shall be unequivocal and faultless.

3. Consistency. The documentation shall not be contradictory to the product description and software product. Every term shall have the same meaning everywhere.

4. Understandability. The documentation shall be understandable for the user who usually performs the stated work task, e.c. by usual business terminology, GUI e.c.

5. Clearness. The documentation shall be easy to review so that the relations are evident. It shall be furnished with the table of contents and index. If the documentation is not available in a printed form, the procedure for its printout shall be included.

3.1.3 Quality requirements for programs and data

The quality requirements for programs and data as well refer to the six basic quality characteristics according to ISO 9126. Unlike the testing procedure where the tester ascertains the consistency of the program functioning (modulus, product) with specifications, the tester of the software product

1. ascertains if the product is functioning as described in the documentation,
2. controls the result correctness in comparison to the known input data, and
3. controls the product functioning consistency with the professional rules, regulations, legislation etc.

Test laboratories are not engaged in the question whether a function is included in the product or not if this is not required by the profession. Equally, they are not engaged with the problem, in what a way an user's interface is made. Test laboratories control the declared facts in comparison with the actual condition, and what is equally important, considering the requirements, good practice, regulations, legislation, profession - the one the product refers to and the informatics.

The requirements for functionality, reliability and usability are clearly defined. But no specific requirements for efficiency, maintainability and portability are defined. However, those product characteristics shall be conformed with the product description and documentation.

3.1.4 Instructions for testing and testing activities

The German standard DIN 66285 and the international standard ISO/IEC 12119:1994 quote in special chapters the testing regulations, how the software product shall be tested. In the regulations, the conditions for the test approach, test execution and test documentation are stated. The regulations require that all the quality requirements have to be tested, as they are defined in the product documentation. It shall be tested as well, if the product fulfils the requirements stated by the profession, good professional practice, legislation, regulations etc. It shall be emphasised that testing is not limited only to the consistency of the product functioning with the documentation, but it shall also include wider professional requirements.

Program and data testing shall be performed on principle of the "black box", and the documentation survey on principle of "inspection".

The test shall be performed in three phases: preparation for testing, test execution, and test report. During the test execution the test record shall be made.
The test laboratory may begin with testing when the following test pre-requisites are fulfilled:
1. The tester shall dispose of the complete software product as supplied by the product supplier. This means the documentation, and all the programs and data, when they exist.
2. The test laboratory shall dispose of the hardware and software as required by the software product.
3. In the case that the education of the user is necessary, the tester shall have access to the educational programs and literature.

Because of the lack of space, only the testing of programs and data is discussed herein. It can be spoken of classical testing of the software product, yet with one remark, that the tester does not correct any inconsistencies between the standard requirements and the test results, but he only records them in the test record. It shall be specially emphasised that all the functions stated in the documentation shall be tested for the functional suitability, in all the stated configurations and above all also in boundary values. Some questions to be responded by the tester are as follows (LINDERMEIER 93 [5]):

- Are all the functions stated in the documentation performed as they are described? This refers particularly to all procedures, masks, functions, data, limit values, efficiency, conformity to regulations, laws etc.

- Are the programs and documentation terminological harmonised? For example the information on faults, commentaries, masks, help functions etc.

- Is the control of the program execution uniform?

- Is the software product reliable and safe (?) in all hardware and software configurations? This requirement is particularly important in the following cases:
    - with boundary burdens
    - when incorrect data are put in
    - with misusage (forbidden usage), etc.

In these cases the product shall not come into the condition that is not understood and mastered by the user. The programs and data shall not get lost, and there shall always be the possibility of reconstruction.

- Do the undefined functions behave equally as the defined ones?

- Are all the product commentaries such that they are understood and suitable answered by the user. This is particularly important for the

information on faults produced by system informations such as: compiler, operation system, DB messages etc.

- Is the product application is such that:
  - the user always knows which function is being executed
  - the masks are clear, understandable and the field length evident
  - the execution is adapted to the professional requirements.

- Are all the functions, representing a data change (deleting, updating, indexing), protected to avoid the loss of data when misused. For instance, confirmation of delete, possibility of reconstruction etc.

As evident, testing is not only classical control of program correctness, but the tester independently, in correspondence with the professional requirements tests the program characteristics such as reliability, usability e.c. These characteristics are usually not stated in the product documentation nor demanded by the standard to be stated. Thus the tester's task is not only to test the syntactic correctness of the programs and data but also to act in part of a very pretentious user and informer and to test the product from these two aspects.

The result of the program and data testing is the record, from which it is evident which standard requirements have been fulfilled and which not. The record shall contain all the data based on which every statement can be verified. All the data and programs that the test has been performed with, shall be protected. All the statements of a subjective character can only be an opinion and shall be stated as such ones.

4. Added value analysis

The practice has shown that a good testing of a software product means a good co-operation between the supplier and the test laboratory. However, this does not mean that the test laboratory shall make allowances to the standard requirements, but that the supplier learns from the testing laboratory about testing principles. And vice versa, that the laboratory learns from the supplier about novelties and innovations, incorporated in the software product. The practice also shows that testing can be efficiently performed only by a tester who knows well the field of action covered by the software product. This means that the business applications can be efficiently tested only by persons with knowledge and experiences from this area. When a certain sphere of action is new to the testing laboratory, it shall provide the necessary knowledge in the form of additional education or it shall hire foreign experts.

The added value to the certificate, obtained by the supplier during the procedure of the software product testing from the testing laboratory is:

1. The supplier learns which are the minimal requirements for a good documentation and data and can include these perceptions into other products.

2. During the process of the certificate acquisition the supplier is forced to test the program and documentation more rigorously. In this way he can eliminate the deficiencies that would otherwise be discovered by the user.

3. The test laboratory transfers its perceptions from other products to the supplier. In this way the ideas and requirements spread. The knowledge, what a quality product is, becomes more and more distinct, the quality characteristics more and more measurable.

4. The supplier shall transfer the requirements of the test laboratory back to the procedure. In this way the testing laboratory with its quality requirements indirectly influences the supplier's procedure and also the quality of all other uncertified products.

5. It is possible that the supplier and the test laboratory agree for the future testing already in the period of the software product elaboration. By our opinion such an inclusion is possible in the phases of the functional program and documentation testing. Although such an inclusion is not directly foreseen in the standard ISO IEC 12119:1994 we are convinced that such a co-operation can essentially reduce the costs of the testing laboratory and means also the flow of knowledge from the test laboratory to the supplier. This co-operation also influences the procedure of the supplier and indirectly all products of this supplier.

The testing laboratory as well acquires a certain added value to the payment of testing - namely the experiences and innovations incorporated into the product by the supplier. They are transferred by the testing laboratory to others and in this way the quality requirements are being increased permanently.

5 Quality requirements and testing limitations

Testing of the software products has some limitations. Let us see the most important ones!

The software products are limited by the standard for program packages. This means that the quality characteristics for information systems in large companies, the process software, produced for each buyer separately etc. are not being stated during testing. Although the standard does not keep distant from the software with potential large economical losses or losses of human lives, we are convinced that the standard requirements for such software products are too modest.

In spite of the defined procedure and criteria testing is still quite subjective. This means that the selection of the test data is left to the tester and that it is impossible to detect all the faults in the programs. This also means that two different testing laboratories will not discover the same faults and that their opinion in limiting cases can be different.

Testing is not cheap! Testing of complex packages can last for months! The result is that the client will think thoroughly before ordering the testing and will do it only if this is demanded by the market! The market demands are the greatest motive for the software producers. If the market does not demand certified products, the producers will not order the testing of their products. The undeveloped markets, such as the Slovene market, do not know this significance yet. The interest for testing is great, yet the producers do not invest into the certification procedure, until the buyers do not distinguish between the certified and other products!

6 Practical experiences in Slovenia

In Slovenia, six year ago a general and world-wide compatible certification schema was started. The basic investment was provided by the government through research projects. The certification body has issued three software product certificates by now. The interest of the small developers is huge, but the demands of the Slovene market are negligible. Therefore small developers do not invest into the certification procedure. They would like to have a certificate, but they wait for a market demand. They also do not invest into MS and therefore only the market requirements improve the quality of their products.

The products tested were typical accounting system products. We are convinced that this kind of products shall be certified because of the legislation requirements. We believe that a huge number of such products (several hundred products for some 7500 companies in Slovenia) will decrease with the Information System Auditing procedures in the next years! The remaining products have to be certified, partly because of the legislation, partly because of market demands.

The product certificate based on the ISO IEC 12119:1994 or DIN 66285 requires that all mandatory requirements are fulfilled. The Slovene certification body controls the proper use of the certificate and a surveillance procedure is organised over the certified product.

The costs of a testing laboratory and the certification procedure for a typical accounting software product are from 6000 to 10.000 DEM in a three-year-period, not including the developer's costs. To our knowledge, the costs to acquire the QMS certificate for a small software house would be from 10.000 to 15.000 DEM, not including the costs for consultants and certification procedure. Therefore it can be expected that small software houses with 5 or

less employees will rather require the product certificate, while larger software houses the MS certificate.

7 Conclusions

I have the privilege to be involved as a testing laboratory manager in product testing and certification procedures, and as TickIT Auditor as consultant in QMS development procedures in several companies in Slovenia. Companies with product certificates agree with all our findings on the added value. One of the managers said: "Once we have got a certificate, we are almost out of business. This is because our product is now practically without errors and we can not charge our errors as maintenance costs to our clients any more! We have to think to sell them something new and innovative!"

The market demand for certified software products is surprisingly small in Slovenia. We are convinced that the open market will rapidly change this situation in special areas, such as accounting systems, tax systems etc. This will be interesting not only for small software houses but also for other developers.

Some requirements of the ISO 9001 QMS standard are without additional resources difficult or even impossible to be met for a small developer. For instance the principle of independence in all kinds of reviews. The QMS procedures, quality records and above all the quality manual tend to be bureaucratic for a small developer.

Our experiences show that the product certification procedure and finally the product certificate can be an alternative approach for a small software house. Product certificates guarantee a better product quality. Testing procedures and surveillance procedures have a positive influence on the developing process. Therefore the ultimate goal - a better software - is reached.

Of course there are limitations to this optimistic statement. They were discussed in paragraph 6. Generally: large and complex software products with a potential economic loss or human life loss are not covered by the ISO/IEC 12119:1994 standard!

REFERENCES

1. DIN 66285:1990. Anwendungssoftware. Gütebedingungen und Püfbestimmungen. Buch Verlag Berlin 1990.

2. ISO/IEC 12119:1994. International Standard. Information Technology - Software packages - Quality requirements and testing. ISO/IEC Geneve 1994.

3. ISO/IEC 9126:1994. International Standard. Information technology - Software product evaluation - Quality characteristics and guidelines for their use. ISO/IEC Geneve 1994.

4. Pivka M. Software quality system in a small software house. Represented at SQM 94 and published in SQM 95 Proceedings.

5. Lindermeier R. Softwarequalität und Softwareprüfung. R. Oldenbourg Verlag München Wien 1993.