

Software quality system in a small software house

M. Pivka

*School of Business and Economics, University of Maribor,
62000 Maribor, Slovenia*

ABSTRACT

The characteristics of a small software house and implementation of ISO 9001 standard in its environment are represented and discussed. Basic thesis, that there is a real need for national or international minimal requirements or levelled requirements for harmonization with ISO 9001 is rejected. In extreme circumstances (few man software house) or for special software products, product standards seem to be more usable than software process standards.

1.DEFINITION OF THE PROBLEM

The software maturity model by SEI (Humphrey¹) anticipates the development environment to be developed from the first, initial level to the fifth, optimized one. The software maturity process from one level to the next one requires a corresponding project organization and management, a well defined and managed process and the software metrics and decision making based on the process data, resources and software products. All the afore stated requires a corresponding level of administration and a certain bureaucracy. Software quality standards such as ISO 9000-3 guidelines (ISO 9000-3²) and IEEE standards presume the management to understand the requirements, and the extent of bureaucracy to be subordinated to minimal management requirements. Because of incorrect understanding of the standard requirements the bureaucracy might become economically questionable, if not already unjustified. Efficient economical implementation of the standard requirements and the software maturity model is a especially burning problem in small software houses. The problem discussed in this paper is the question, how to introduce the quality system into a small software house in an economical and efficient way.

84 Software Quality Management

2. BASIC THESIS

More and more often software purchasers, such as the state, telecommunication firms, railway, post, banks, insurance companies etc. require from the software suppliers the evidence of having implemented the software quality system. Other buyers of the software, such as larger or smaller companies, are also aware of the signification of the supplier's quality. The software purchasers and users often require several certificates and assurance for the supplied software to be in conformity with general branch rules, with various standards, declared functions etc.

The characteristics of post-communist European countries is the downsizing of large enterprises and reconstruction thereof into small and medium ones, as well as foundation of numerous new small firms. Most often, the information centers in large firms have been transformed into new small software houses. In Slovenia, with only about 2.000.000 inhabitants, there are several hundred software houses, elaborating the majority of the new software being used in Slovenia nowadays. The two-years experiences in introducing the DIN 66285 standard in Slovenia (DIN 66285³, Pivka⁴) and the first practical results of the quality system initiation awoke doubts in the possibility of a rapid implementation of quality standards in small software houses.

Our theses are as follows:

1. A rapid implementation of the software quality system as instructed by ISO 9000-3 guidelines or TickIT scheme (TickIT⁵) can be economically questionable for a small software house.
2. The elaboration and introduction of their own in-house software quality system is doubtlessly necessary for a small software house, yet not recognized by the purchasers, as they trust only the international standards.
3. From the first two theses proceeds a further one, that there is an actual need to establish the national or international minimal quality system requirements for small software houses, which would be in harmonization with standards such as ISO 9001⁶.
4. For some small software houses it is economically more justified to obtain the product certificate than the software system certificate.

3. CHARACTERISTICS OF A SMALL SOFTWARE HOUSE

Speaking about small software houses is rather questionable, as a generally recognized definition of a small software house is not known. It would be nice, if software houses were classified by the number of their employees and thus small software houses were those having for instance less than 10 employees, medium software houses those with 10 to 30 employees and

large software houses those with more than 30 employees. Because of obvious reasons such a classification is not possible. We suggest to describe a small software house by its specific characteristics. By our opinion they are as follows:

1. Software product size. Small software houses develop and maintain small and medium software products. The size of a software product can be expressed in different terms: by the number of source code lines, by the number of the functional points, by the number of the documentation pages and by other metrics. A very simple and useful one is the metrics of source code lines and the documentation extent. For this metrics the SCOPE (SCOPE ⁷) project suggests the following distribution of software products:

size	example	KLOC	# of pages of doc
small	embedded type applications	< = 10	cca 50
medium	PC-sized application	cca 50	cca 250
large	workstation mainframe aplic.	>= 100	cca 500

Table 1: Software product size

2. The number of users for whom the software house maintains the software. By our opinion the number of a small software house users depends on the following:

- quality of the products;
- sales policy, e.c. price;
- software product size;
- software house productivity;
- software tools used for the software development;
- application types;
- hardware for which the software product is intended for;
- competition at the market;
- etc.

Of course, each of the above stated dependences has a different influence on the number of a software house users. By our experiences it can be concluded, that for typical PC applications, classified to medium software product according Table 1, the number of users of successfull small software houses should be from 50 to 150.

3. The number of employees in a software house. The number of employees in a software house certainly depends on several factors. The most important are the following ones:



86 Software Quality Management

- the number of users;
- the capability (productivity) of the employees;
- the maturity level according to the software maturity model;
- software tools (workbench, programming languages, company tools, test tools...);
- application type;

Based on the data known for Slovenia it can be concluded, that the number of employees for typical medium size PC software products depending on the number of users is as follows:

Number of users	Number of employees
cca 20	1 to 2
from 20 to 50	3 to 5
from 50 to 100	5 to 10
> 100	> 10

Table 2: Number of users, number of employees

4. In small software houses the software process dynamics is considerable. The work is not divided among several people, and each one is responsible for several fields of work. That means that employees have to be masters of various knowledges, specialization is difficult. The software development process is not carried out by some of well known models (V model, prototyping, waterfall) or methodologies (structured methodologies), but only in two, three phases not being clearly separated: design, coding and implementation. Very often all phases are carried out by the same person. Phase results verification is very difficult if not quite impossible.

5. Small firms require a high level of the mutual confidence. If the main designer (of course this is not possible if he or she is also the owner of the firm) or the programmer leaves the firm, this can cause great troubles for the house.

6. In most cases, small software houses develop highly specialized software products. This also requires the development of in-house process models to establish the software and development of in-house tools. Small software houses use classical compilers, data bases and operation system only as a foundation to be built up for their own needs. The software house, establishing its own developing tools (for instance in the object oriented environment), in most cases do not think about CASE tools and other up-to-date technologies. Such a software house looks for the most economical way in the software development process. The new technologies are carefully selected, and used only when they are proved to be economical.

4. SOFTWARE QUALITY SYSTEM BASED ON ISO 9001 AND ISO 9000-3 GUIDELINES

The quality system as required by ISO 9000 international standards proceed from the following fundamental recognitions, being of essential significance for the development of products of good quality:

- commitment and responsibility of the management;
- developer's and user's responsibility;
- supplier should establish and maintain a documented quality system (ISO 9000-3, paragraph 4.2.1);
- constant, independent control of the results obtained;
- the development should be organized in accordance with a specified model;
- it is essential to plan the development and maintenance of the quality system;
- efficient execution of supporting activities, as configuration management, document control, process, product and resource measurements, subcontractor evaluations etc. is very important.

This means that the quality system shall define the organization, allocate the responsibilities and authorizations, and define the process, resources and procedures, assuring the implementation of the process within the planned terms, resources and quality. The quality system consists of the following: organization structure of the firm, allocated responsibilities, a defined process and procedures for carrying out the process, and resources. The most important message of the standard is, that only the process constructed according to the above stated presumptions can effectively be managed.

Regarding all their characteristics, small software houses usually response negatively to the quality systems according to ISO 9001, and ISO 9000-3 Guidelines, respectively. There are several reasons for it: from the unacquaintance with the standard requirements to the fact that by their opinion the requirements are too bureaucratic and they only lower the productivity.

Further on, the key areas of the ISO 9000-3 Guidelines are analysed, as well as the possibilities for the efficient implementation thereof in small software houses.

4.1 Quality system - Framework

According to ISO 9000-3 Guidelines the quality system shall be arranged in such a way that it clearly defines the following: quality policy, responsibilities and authorities of the performer, user's responsibilities, carrying out the principle of the independent control and verification in all process phases, and the procedures and types of process and quality planning. For a small

88 Software Quality Management

software house with only a few employees, say 2 or 3, it is rather questionable to perform such a quality system to make possible the principle of the independent control. Phrase 4.1.2.2 - Verification resources and personnel of ISO 9001 Standard namely requires the following:

The supplier shall identify in-house verification requirements, provide adequate resources and assign trained personnel for verification activities. Verification activities shall include inspection, test and monitoring of the design, production, installation and servicing processes and/or product; design reviews and audits of the quality system, processes and/or product shall be carried out by personnel independent of those having direct responsibility to the work being performed.

The first problem to be solved by a small software house if they want to established their own quality system according to the requirements of the ISO 9000 standards is, how to implement the principle of independent testing, control, verification and internal audit in all process phases. In extreme cases, i.e. when there is only one person in the firm, this principle is impossible to be implemented without additional resources. Yet also in the case when there are 3 to 5 employees in the firm, the implementation of the principle is possible only formally, not actually. The problem is the independence and unsufficient resources to carry out all the required reviews.

By our opinion the problem of independence and sufficient resources can be solved in one of the following ways, that are not contrary to the requirements of the above stated standards:

1. The firm may hire one or two consultants, to take in the quality system the part of the independent auditors in individual pocess phases and who perform periodic control of the efficient quality system implementation.
2. The firm may organize one or two work teams to perform control reviews in all process phases: from the contract review to the result verification. In the case of only one work team in the firm the team members would perform control reciprocally. Yet such a control can hardly assure the principle of independence. It is a matter of fact, that a designer can not control the programming practice and vice versa. By such implementation of the independence principle the standard requirements are formally met, yet not actually.
3. Irrespectively of the number of employees a small software house may periodically hire consultants for reviewing the quality of process, products and the efficiency of the quality system implementation. The firm itself would define the control points of the process and the frequency of quality system reviewing.

Of course the firm itself will make decision which of the stated possibilities to choose. Proceeding from the aspect of the standard requirements, we think

that the optimal combination of including the consultants is the one that is economically justified and at the same time assures independent verification and auditing.

4.2 Development planning

ISO 9001 Standard and the ISO 9000-3 Guidelines devote a whole chapter to the problem of development planning (chapt. 4.4 and 5.4). The significance of this chapter lies in the fact that it is impossible to manage the processes in the programme engineering if they are chaotic, intuitive, without resources and left to individuals and their intuitive decisions. Thus not without reason the directives of the chapter 5.4.1 of ISO 9000-3 Guidelines require the following:

The development plan should cover the following:

- a) the definition of the project, including a statement of its objectives and with reference to related purchaser or supplier projects;*
- b) the organization of the project resources, including the team structure, responsibilities, use of sub-contractors and material resources to be used;*
- c) development phases (as defined in 5.4.2.1);*
- d) the project schedule identifying the tasks to be performed, the resources and time required for each and any interrelationships between tasks;*
- e) identification of related plans, such as*
 - quality plan,*
 - configuration management plan,*
 - integration plan,*
 - test plan.*

The development plan should be updated as development progresses and each phase should be defined as in 5.4.2.1 before activities in that phase are started. It should be reviewed and approved before execution.

That means that development planning is an essential aspect of software production. Key areas of the development planning are as follows:

- phased development,
- use of methods and tools,
- progress control,
- input / output control,
- verification.

For each of these key areas the quality system should anticipate, how to plan it.

1. Planning phased development. The software process development is a project-based process. Typical for a small software house is a small

90 Software Quality Management

number of phases. Therefore a good project definition can be a plan meeting the standard requirements. To meet the formal standard requirements from the aspect of planning and to satisfy the needs of a small software house the project definition should minimally include the following:

- required phases to be carried out by display of the WBS structure or in some other form. It is important that the project definition clearly shows the path, how the user's requirements are to be mapped in the expected software product;
- required inputs and outputs;
- necessary resources (personnel and monies);
- the time foreseen to carry out individual phases;
- the way foreseen to verify the results of individual phases.

2. Methods and tools. The quality system should identify methods and tools to be used in the software development. A small software house usually does not use any of well known commercially available methods such as SSADM, Jackson, Yourdon, etc, but develops the software only in the following phases: design, programming and testing, installation and maintenance. Work method planning is simple and meets the standard requirements only in the case when these methods are satisfactorily documented in the quality system. The standard orders neither the methods nor the tools. Yet it is required that the following is to be documented in the quality system and controllable in the practice:

a. Method of the software production. For a small software house it is enough if the phases and work methods are described. If in the design phase a small software house works according to some well known methods, e.c. structural methods, the standard requirements are fulfilled if the references are listed. The same shall be considered for programming and testing. For programming the programming rules, internal tools etc. shall be prepared or made available in some other way.

b. Training, references, manuals and similar for the selected method and tools. The everyday practice uses various tools, from in-house developed tools and procedures to complex integrated workbench design and development tools. Small software houses seldom use workbench design and development tools, they rather develop in-house specialized tools, or they purchase foreign products to integrate their own development therein. For all in-house and foreign tools or foreign software the quality system shall document the training and literature and anticipate the in-house verification as well.

c. Purchase and verification procedures for tools and foreign software products. The procedure shall provide the correct selection, arrangement of usage licences and similar.

3. Progress and input / output control. Efficient management requires the review of the process performance. Irrespective of the fact, that in a small

software house the responsibilities are well known, it is nevertheless necessary to plan the progress review. The progress review is performed in various project phases. The purpose of review is to identify the differences between the actual and planned project state. In a small software house at least the following control points shall be planned:

- contract planning review where the user has to participate completely,
- contract holding reviews,
- process and resources control.

4. Verification. The software development planning shall cover verification of all activity results. Verification planning is a procedure by which the software house defines the methods to be used for performing the verification. For a small software house various methods in various development phases, such as structured walkthroughs, Fagan Inspections and similar, are unfeasible as for time and personnel. This does not mean that a small software house could not plan the verification. An efficient process management requires verification, yet in a small software house this can be done only in terms of controls after individual process phases, based on in-house work methods. For example: design review, test results review, good programming practice review and similar.

4.3 Quality planning

The software quality shall be planned as well. The product quality can be planned if certain input / output criteria of individual development phases are available. The quality system shall thus have documented results of individual phases and measurable criteria, by which these results will be verified. So the quality plan consists of one or more documents, documenting the quality objectives, input / output criteria and the methods to review the obtained partial results and the final result in a most measurable way.

For a small software house it is sufficient to plan the quality by defining as much as possible measurable input output criteria for each development phase and by planning the review methods. In the cases of uncritical software products or small projects, the constituent parts of the quality planning can also be the following: configuration management plan, quality records, collecting process and product data, purchasing procedures and training.

4.4 Design and implementation

In the design and implementation phases, user's requirements are transformed into the software product. The work methods are identified in the project plan and defined in the quality system. Also in a small software

92 Software Quality Management

house, the software quality system should be performed in such a way as to provide a good software product on the base of a well defined and managed process and not on the base of longlasting and expensive testing and evaluation.

By our opinion the quality system in a small software house and its execution in the design and performance phases according to ISO 9001 standard and ISO 9000-3 Guidelines should represent no problem. There are no standard requirements that could not be met in a small software house due to limited resources. When for instance, the rules of good programming, naming conventions etc. are defined in the quality system, then the individual performers are responsible for considering and improving these rules, yet the control and verification system shall prevent the deviations thereof and provide the transfer of knowledge and experiences.

In the practice the problem of small teams (extremely: one man band) is not so much in defining the quality system but in the well disciplined execution according to the accepted rules. As it is difficult to assure the principle of the independent control, the discipline as well depends on the individual persons. This problem has already been analysed in the introduction. We suggest once again to include an independent consultant at least in the key verification points of the design and execution.

4.5 Testing and validation

The software engineering knows various types of testing, from testing a few code lines to the whole software product testing. It can be performed in different ways, several strategies are well known. The ISO 9001 standard, and especially ISO 9000-3 Guidelines devote every attention to testing. Therefore also in quality systems of small software houses at least the following test related points shall be well defined:

- testing strategy, depending at least on the programming language and the software product criticalness,
- testing methods for individual programs, program groups and the complete software product,
- documentation related to test data and test results,
- activities related to the test results validation,
- testing resources: personnel, test tools etc.

For a small software house these requirements represent a considerable problem. A specially heavy burden is the independent preparation of test data and documentation of test results. This is specially true for uncritical products, such as typical PC applications. The standard test requirements are considered as unnecessary bureaucracy, making the process expensive and unefficient.

It is characteristic for a small software house that the test data preparation and testing is performed by software item authors. Principally this diminishes the quality of testing. A software item author tests the correctness of the subjects he has performed by himself and not the subjects he should perform. The author might have misunderstood the specification, the specification could have been wrong, deficient or even undocumented. This problem can be solved only in such a way, that a small software house invites the user or an independent specialist as his representative to participate in the testing procedure as much as possible. The user or his representative may be included contractually or in some other way. This person would prepare or at least verify the test data and should be included in the result validation procedure as much as possible.

4.6 Acceptance

By the standard requirement the purchaser and supplier shall mutually agree, in what way the purchaser should judge whether or not the supplied product is produced in accordance with the previously agreed criteria. The standard requirements order the purchaser and supplier to plan the acceptance mutually and anticipate the required activities. In the quality system at least the following activities, being the constituent part of the contract shall be planned:

- acceptance schedule,
- acceptance object (not only programmes but also documentation, procedure, data, training,.....
- hardware and software facilities for acceptance process,
- acceptance criteria.

The stated activities, that have to be defined in the quality system and included in the contract, should not represent any problem for a small software house. Irrespective of the number of employees and the criticalness of the product the purchaser always accepts the product in some way. The standard requires only that the acceptance should not be made by chance. This means that all the evidenced problems or deviations from user's expectations shall be solved in a suitable way.

4.7 Replication, delivery and installation

The requirements on replication, delivery and installation of the software product are clearly defined by ISO 9000-3 Guidelines. The definition of these conditions in the quality system of a small software house represents no problems. It depends on the software product type, hardware and software environment, licences, anti-virus protection etc. The standard does not require the replication, delivery and installation conditions to be the constituent part of the quality system. By our opinion it is sufficient if these conditions are clearly stated in the contract.

94 Software Quality Management

4.8 Maintenance

The purchasers are still not aware of the fact that the purchase of the software product represents only the minor part of all costs in its lifetime. A much more expensive part is maintenance which does not mean only the correction of errors but first of all functional expansion of the software product and its adaptation to new user needs and to changes in the user's environment. The ISO 9000-3 Guidelines devote every attention to maintenance. The essential requirement is that maintenance shall be considered equally as development. This means that the maintenance procedure shall be defined in the quality system. This holds true also for small software houses. In small software houses maintenance can be organized even more efficiently than in great ones. Maintenance is namely performed by software items authors and this is not contrary to the principle of independence.

When maintenance of the software product is required by the purchaser, the maintenance procedure and conditions shall be stipulated in the contract. Yet there is a question whether the whole maintenance procedure shall be defined in the quality system as well. As the supplier may maintain various software items under various conditions, we suggest for a small software house to document in its quality system, what services are offered to the purchaser, and to stipulate them in detail in the contract. We suggest to define the following in the quality system:

1. Maintenance life cycle for individual maintenance types. There are not the same life cycle activities for correcting the errors, general functional expansion or interface changes.
2. Procedures of maintenance planning and providing the maintenance resources.
3. Documenting procedures for activities and maintenance results. Very important is the problem documentation (product metrics!) and the problem resolution analysis (ISO 9000-3, paragraph 5.10.6).

All these procedures shall be formalized in the maintenance contract, when maintenance is required by the purchaser after initial delivery. According to standard requirements the software house shall define the maintenance procedure in its quality system even if it has only one person employed. In each and every contract concluded with the purchaser the individual quality system regulations are operationalized depending on the product and maintenance type.

4.9 Quality system - supporting activities

According to ISO 9000-3 Guidelines all phase independent activities are counted to supporting activities. They are: Configuration management,

Document control, Quality records, Measurement, Rules, practices and conventions, Purchasing, Included software product and Training.

1. Configuration management. When a small software house develops uncritical software products, the management of software product versions may be a constituent part of the quality plan. Therein should be defined what a product version is, how do various versions affect each other, how are the version changes to be carried out, and how to document the version changes. It is essential that the control of every software item in all lifetime cycle phases is documented. A small software house can not avoid these standard requirements. It has to realize them in the quality system in the way and scope as necessary.

2. Document control. Generally, the documentation is a weak point of the software engineering. This holds specially true for small software houses, always lacking time for "paper work", as the purchaser requires products and not papers. The standard is distinct as for the documentation, therefore a small software house has to include all the procedures for the document issue, document usage control and document withdrawal. This holds true for all project documents: from contracts and control reviews to the final products of individual activities in the process. The standard does not require all the documents to be written on paper, they may be presented on any electronic or other medium. It is sensefull that a software house elaborates a minimum of the required documents, still assuring the efficient procedure and product documentation.

3. Quality records. *The supplier shall establish and maintain procedures for identification, collection, indexing, filing, storage, maintenance and disposition of quality records (ISO 9001, paragraph 4.16, first section).* A small software house may define the quality records and its usage and maintenance in the quality plan for each project, or may define them in the quality system. The standard requirements are distinct, therefore a small software house has to search after the optimum between the standard requirements and actual needs.

4. Measurement. In the software engineering the process, product and resource metrics are known. The standard does not order any metrics, for there is no universally aknowledged metrics known. Yet it is anticipated that for the development and maintenance management certain process and product measurements are necessary. The purpose of the standard requirements for metrics is not the popularization of progam metrics but collecting the management data to attain better quality. A small software house as well needs some minimal process and product metrics. Their purpose, method of data collecting and operating procedures may be a constituent part of the quality plan or quality system.

5. Rules, practices and conventions. According to paragraph 6.5 of ISO 9000-3 Guidelines the supplier should assure efficient implementation of the required quality system. There are rules, practices and conventions on how

96 Software Quality Management

to make the quality system effective. For a small software house these requirements are rather trivial as the quality system continuously confirms itself during the verification, validation and control reviews procedure. By our opinion no special requirements are necessary for an efficient performance of the quality system in a small software house. These requirements are implicitly included in the quality system management.

6. Tools and techniques. The supplier shall assure that all tools and techniques used are carefully selected and applied in the practice. This holds true for small software houses as well. They seldom purchase complex tools such as CASE workbench. They mostly develop inhouse tools, which requires corresponding documentation and training.

7. Purchasing. The software houses purchase not only the software as compilers, spreadsheet programs, but also tools and hardware. By the standard requirement it shall be evident for each delivery that the selected product or subcontractor conform to specified requirements. This means that the quality system shall include the purchase procedure for hardware or software. By our opinion it is sufficient for a small software house if the purchase procedure is documented and it shall be evident from this documentation that the specified criteria have been fulfilled.

8. Included software product. Sometimes the supplier may include or use a foreign, maybe purchaser's software product. According to standard requirements this software product shall be suitably selected, stored and protected. A small software house does not need any special rules and procedures to be defined in the quality system for such a case. The conclusion of a contract with the subcontractor is sufficient and the quality criteria, licence clauses and other specific requirements shall be defined therein.

9. Training. Although it is clear that knowledge is imperative for efficient work, small software houses invest very little in training. According to standard requirements the supplier should identify the training needs, test the knowledge level and organize training if necessary. Small software houses can not avoid this requirement as well.

5. CONCLUSIONS

In this paper the requirements and possibilities of introducing the ISO 9001 standard in small software houses are discussed. Our conclusions on the theses stated in the introduction are as follows:

1. thesis: A rapid implementation of the software quality system as instructed by ISO 9000-3 Guidelines can be economically questionable for a small software house.

Conclusions: A small software house has to be on the third level according to the Software Maturity Model as well if it wants to have an established quality system according to ISO 9001 standard. For countries as Slovenia, where the great majority of software houses is still on the fundamental level, introduction of ISO standards means first of all settling of the management problems, and only after that they may start solving the organizational problems. In software houses with a weak management not understanding the maturity process, the decision for introducing the ISO 9001 standard is wrong. There is a high probability the initiated process to be interrupted after a certain period of time. The reason therefor lies in the fact that the software house expects results not being realistic for its maturity level.

2. thesis: The elaboration and introduction of their own in-house system is doubtlessly necessary for a small software house, yet not recognized by the purchasers, as they trust only the international standards.

Conclusions: The problem of acknowledging the supplier's quality system by the purchaser is a multilevel one and could not be analysed directly in this paper. The problem lies in the market development and purchasers' requirements. Simply said, if the market is more developed, there are more needs for certified quality systems.

3. thesis: From the first two theses proceeds a further one, that there is an actual need to establish the national or international minimal quality system requirements for small software houses, which would be in harmonization with standards such as ISO 9001.

Conclusions: The analysis of the ISO 9001 standard showed that actually there are only a few requirements to be considered as bureaucratic ones and difficult for a small software house or even impossible to be met. First of all this is the principle of independence in the review and verification procedures. This principle can be solved only by additional resources, which may be economically questionable. Another problem, implicitly presented in the analysis, is that a software house does not understand the standard requirements and defines the process so much in detail that the execution becomes bureaucratic and unefficient. Irrespective of both problems by our opinion it can not be expected that national or even international minimal quality system requirements for small software houses would be elaborated.

4. thesis: For some small software houses it is economically more justified to obtain the product certificate than the software system certificate.

Conclusion: In Slovenia, the procedure of acquiring a conformity certificate according to the German standard DIN 66285 costs from 6000 DEM to 10.000 DEM in three years for average PC sized products. The procedure to acquire the software system certificate costs a small software house in Slovenia from 10.000 to 15.000 DEM. Both prices are true for the circumstances in Slovenia, without inclusion of consultants. Therefore it can be expected that small software houses with less than 5 employees would rather require the product certificates, while larger software houses the



software system certificates. If in other countries the testing procedure of the software product is more expensive, then this conclusion is not true for such countries.

6. REFERENCES

1. Humphrey Watts S. Managing the Software Process. Addison-Wesley 1990.
2. ISO 9000-3. Quality management and quality assurance standards. Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software.
3. DIN 66285. Anwendungssoftware Guetebedingungen und Pruefbestimmungen. DIN 1990.
4. Pivka M, Borko M. Survey of Software practices and software quality assesment in Slovenia. 3.rd European Conference on Software Quality. Madrid, November 1992.
5. TickIT. Guide to Software Quality Management System Construction and Certification using EN 29001. DTI, 1992.
6. ISO 9001. ISO 9001 1987. Quality systems - Model for Quality assurance in design/development, production, installation and servicing.
7. SCOPE. CEC Esprit Project 2151 SCOPE. Software Assesment and Certificatio Programme Europe.