Marjan Pivka
Vojko Potočan
School of Business and Economics
Maribor
E-mail: Pivka@uni-mb-si

UDC: 681.32.06
Original Scientific Paper

How can Software Packages Certification Improve Software Process

Popular software assesment models such as CMM, BOOTSTRAP, SPICE or ISO 9000 ignore the impact of software product certification on software quality. The first standard for software product quality was German DIN 66285. Based on this standard, the ISO developed a international standard for quality requirements and testing procedures for software packages: ISO/IEC 12119. This paper presents our experience with classical testing models based on ISO/IEC 12119 and DIN 66285 and with our improved model, call ed the cooperative model of testing. With this model we can achieve improvements in testing of efficiency and software production process. Practical experience shows that the roles of all parties in the testing process have changed. Knowledge transfer between the testing laboratory and supplier depends on supplier's maturity level.

Key words: software quality, testing models, ISO/IEC 12119, cooperative model of testing.

1. Problem definition

Assessment, evaluation and the software process improvement are nowadays known through the American CMM model, the European BOOTSTRAP model, the international SIPICE and the ISO 9000 model.

All the models are based on a widely known fact that better product quality can only be achieved by improving the software process. The above stated models do not assure that the software, which is produced in a software process environment and is, for instance, at the third maturity level, will conform with the specific requirements of legislation, safety standards, etc.

The users of such software products need more frequent the confirmation that the program product conforms with specific regulations and legislation and accordingly meets certain safety requirements, standards, etc. Such program products are business software packages, embedded software, safety critical or life critical software, etc.

The testing of the software product activity for conformity with legislation or other specific requirements is only possible if there are standards or regulations for their testing. The assessment models do not provide direct response to the problem even if the program product is in accordance with the specific requirements of the user.

There are more than two hundred software quality standards. Most of them are National or Multinational such as ANSI, BSI and DIN standards, and professional standards such as IEEE Standards, Defence standards etc. They cover software entities such as Management, Quality Assurance, Configuration Management, Safety, Design, Requirements Specification, Coding, Verification & Validation, etc. Only two international standards consider the software product. All the others are a part of the software design, implementation and maintenance process. The two international standards applicable to software products are: ISO/IEC 12119:1994 Information Technology - Software packages - Quality requirements and testing and ISO/IEC 9126: 1991 Information Technology - Software product evaluation - Quality characteristics and guidelines for their use.

The testing of a software product for conformity with legislation or specific regulations undoubtedly has an impact on the software developement process and thus, in the long run, on the supplier's software process improvement.

This paper discusses the testing model of the software products with which we do not only test the software product's conformity with some standard, but also achieve software process improvement with of knowledge transfered from the testing laboratory to the supplier.

2. ISO/IEC 12119

The scope of ISO/IEC 12119: This International standard is applicable to software packages. Examples are text processors, spread-sheets, data base programs, graphic packages, programs for technical or scientific functions and utility programs (such as programs for data administration). It establishes requirements for software packages (quality requirements) and instructions on how to test a software package against these requirements (instructions for testing, in particular for third party testing). It deals only with software packages as they are offered and delivered. It does not deal with their production process (including activities and intermediate products, e.g. specifications). The supplier's quality system is outside the scope of this international standard. (ISO/IEC 12119).

Sets of quality requirements based on this standard are:

1. Product description requirements. Each software package shall have a well-defined product description. It is a part of the software package documentation. It provides information on the user's documentation, programs and data, if any exist.

2. Each software package shall have complete, correct and consistent user's documentation. The documentation shall include complete information on how to install, use and maintain the software package.

3. Programs and data. Quality requirements consider six software quality characteristics defined in ISO/IEC 9126 standard. For functionality, to name one example, it defines the following requirements: installation, presence of functions, correctness and consistency.

ISO/IEC 12119 provides the instructions for the testing of software packages. The specification of the software product and documentation are tested by the inspection method while the testing of programs and data is conducted by black box testing.

The classical mode of such testing presupposes that the program product has been completed, and that the testing laboratory is testing the product which is ready for the market or is already on the market. The outcome of such testing is a test report, which shows whether the program product meets the requirements of ISO/IEC 12119 and all the regulations related to this product.

In such testing, the following facts have been established in practice:

◊ the testing laboratory shall have to elaborate the testing data for testing, which makes the testing procedure demanding in terms of time and costs;

◊ the testing laboratory shall be familiar with the field covered by the software product. For instance, construction industry packages can be tested only by teams that include experts from construction industry. This assures a high quality level of testing not, only of the programs and data but also of the user interfaces and documentation;

◊ the testing laboratory has no direct impact on the developement process of the software product considered;

◊ the knowledge transfer from the testing laboratory to the supplier is limited;

◊ the market (on the users' and suppliers' side) is influenced by ISO 9000 standards and other assessment models and therefore does not understand or accept the meaning of the certified software products;

◊ universities, research laboratories, etc. do not pay attention to the software product certification process.

Due to the above-mentioned reasons the classical model of testing software packages for conformity with ISO/IEC 12119 requirements is not used in practice as often as the above mentioned assessment models. In our opinion, the reasons for this lie in the classical model of testing which requires the already performed work (integral testing and the testing of the documentation conformity) to be carried out by the supplier (supposition) and by the testing laboratory (obligation). The solution can be seen in the cooperative model of testing, which will be presented further on in this paper.

3. Cooperative Model of testing

The characteristics of the cooperative model of testing can be summarized as follows:

The cooperative model of testing is not in contrast with the requirements of the international ISO/IEC 12119 standard. It is used to test the conformity of software package with the requirements of standards, being the basis for the evaluation of software packages. A new value of this model can be seen in its impact on the quality of the supplier's software process.

Testing in the testing laboratory is not an isolated activity, but the one included in the supplier's software process. Such testing, therefore, presents one of the phases in the life-cycle of the software process.

Testing is implemented in many phases, providing a fast reaction to the established deviations. With updated corrections we can reduce the required time and the scope of work, which enables more economical testing.

The testing laboratory, the supplier and the users or any other participants interested in software quality (professional associations, auditing agencies, consulting firms, ...) actively participate in its implementation. The quality of testing depends upon the organizational maturity level, the participants in testing and the know-how transfer between them.

The individual participants of testing have a specific knowledge. The testing laboratory passes the knowledge on the method of testing to suppliers and users. The supplier passes on the knowledge on technology and technique of the software production to the testing laboratory and to the users. The users or other participants in testing pass on specific expert knowledge and practical working experience to the supplier and the testing laboratory.

In the process of the know-how transfer, the cooperative cooperation of the participants in testing assure synergic results which affect the economy and the quality of testing as well as the software production process.

The cooperative model of testing is shown in fig. 1.

Cooperative testing involves preparation works, central phase - direct testing and final works.

The starting points and commissioner basis for the performance of testing are defined in the preliminary phase. The communissioner of testing (supplier, user, professional associations, ...) and the testing laboratory agree upon the purpose, goals and requirements, define the mode (selection of procedure, scope, participants in testing) and the terms of testing (testing plans, testing data, testing documentation, ...). In special cases, the users with demands for testing and determination of particular and desired goals of testing can also take part in preparation.

The central phase of testing follows. The testing of individual components of the software package (unit test) is usually carried out by the supplier without direct supervision of the testing laboratory or the users. In the testing laboratory and on the basis of the development documentation, the supplier examines the correctness and adequacy of individual software components. The testing laboratory or other participants, who pass on specific requirements, knowledge and experience to the supplier, can also take part in the unit test.

After the testing of components we continue with the integrated testing. The integral testing is carried out by the supplier based on the requirements and conditions of the testing laboratory. The testing laboratory carries out the function of controlling supplier's work directly, by control over the supplier, or indirectly, by examining the testing documentation. In the know-how transfer the testing laboratory has an impact on the improvement of the supplier's work, which enables a better quality of current and future products. With their experience and proposals the other participants can have an influence on the integration test.

The central phase is completed with the final testing performed by the testing laboratory on the basis of the standard requirements and contractually - agreed obligations. The work covers the methodological, content and formal assessment of suitability. The selected functions, limited values, user's interfaces, user's documentation, volume, etc. are tested at random.

The testing laboratory can also include the users or other participants in its work to ensure conformity of the software operation to needs and requirements of the user's environment.

In the final phase, the testing laboratory elaborates the testing report, which is the basis for decision-making by the Certification Body (certificate approval, refusal of the approval). At the supplier's special request the testing laboratory can also take into consideration other proposals and opinions.

4. Application of the cooperative Model of testing

Our knowledge and practical experience acquired through the application of the cooperative model of testing can be classified into three groups:

First: the target software program is in process

The reality concerning the planning of testing depends upon the supplier's software maturity level. In the process of development, the program package has been constantly changing, which requires continuous adjustment of testing plans as well as testing both by the supplier and the testing laboratory. Because of development, the documentation is incomplete, there are no sufficient testing data available or as such can not provide real and comprehensive testing of the software package. It was established that the low maturity level of the supplier's software process demands an intensified engagement of the testing laboratory which assures the adequate level of testing with increased control along with the know-how transfer.

Second: testing implementation

The second group of knowledge is related to the implementation of testing and to the configuration management of the software package in process (development). In the practical work has provided insight into the classic problems of testing without the configuration management in testing of the already tested modules; the same errors occur in various version of modules; the documentation does not follow the process (the development) and testing data are not available in sufficient scope and quality.

Third: cooperation and communication

The third group comprises knowledge on the cooperation and communication between the participants involved in the testing. Between the testing laboratory and the supplier a relationship of mutual respect and confidence should exist. Only in this way the reaction to the established deviations can be prompt, less conflicting and does not depend upon redundant formalities. In testing, we may have a situation in which the a program error can not be exactly documented. In such situations the solution is only possible by mutual cooperation and confidence. Also, the relationship between the testing laboratory and the users (potential and interested users) is very important. Their know-how, requirements and wishes represent considerable contribution to the quality of testing.

5. Conclusions

This cooperative model presents a systemic approach to understanding of the software life-cycle as complete process which includes all the operations of the introductory (preliminary, initial) investigations since the termination of its application. Its application enables the knowledge major co-dependency and the synergical operation of the individual activity in the process, which is also true for testing.

In the cooperative model, the roles of the individual participants in testing and their mutual relations are based on creative cooperation and mutual exchange of know-how.

The value of the cooperative model can be entirely seen in the testing of the software package in process (development). It was established in practice that the supplier's software process maturity level presents the most important problem.

Organizational immaturity can be seen as the inadequate definition of the supplier's software life-cycle. For this reason, the supplier is not in a position to adequately carry out and document all development phases, especially the phases of testing. Technological development is usually above than the organizational (e.g. the application of 4GL) but without corresponding organizational support it does not enable the achievement of the expected results.

Our practical knowledge can be summed up as follows:

In the preparation phase, the testing laboratory assesses the supplier's software process maturity level. This can be established by the second party audit of supplier with a selected model (ISO 9001, CMM, BOOTSTRAP, SPICE or some other).

The cooperation mode and control over the supplier depends on their maturity level. If the supplier is at the initial maturity level (CMM model: initial level) a larger scope of know-how and work transfer of the testing laboratory is necessary. The know-how transfer is related to the life-cycle, project management, configuration management, methods of testing, etc.

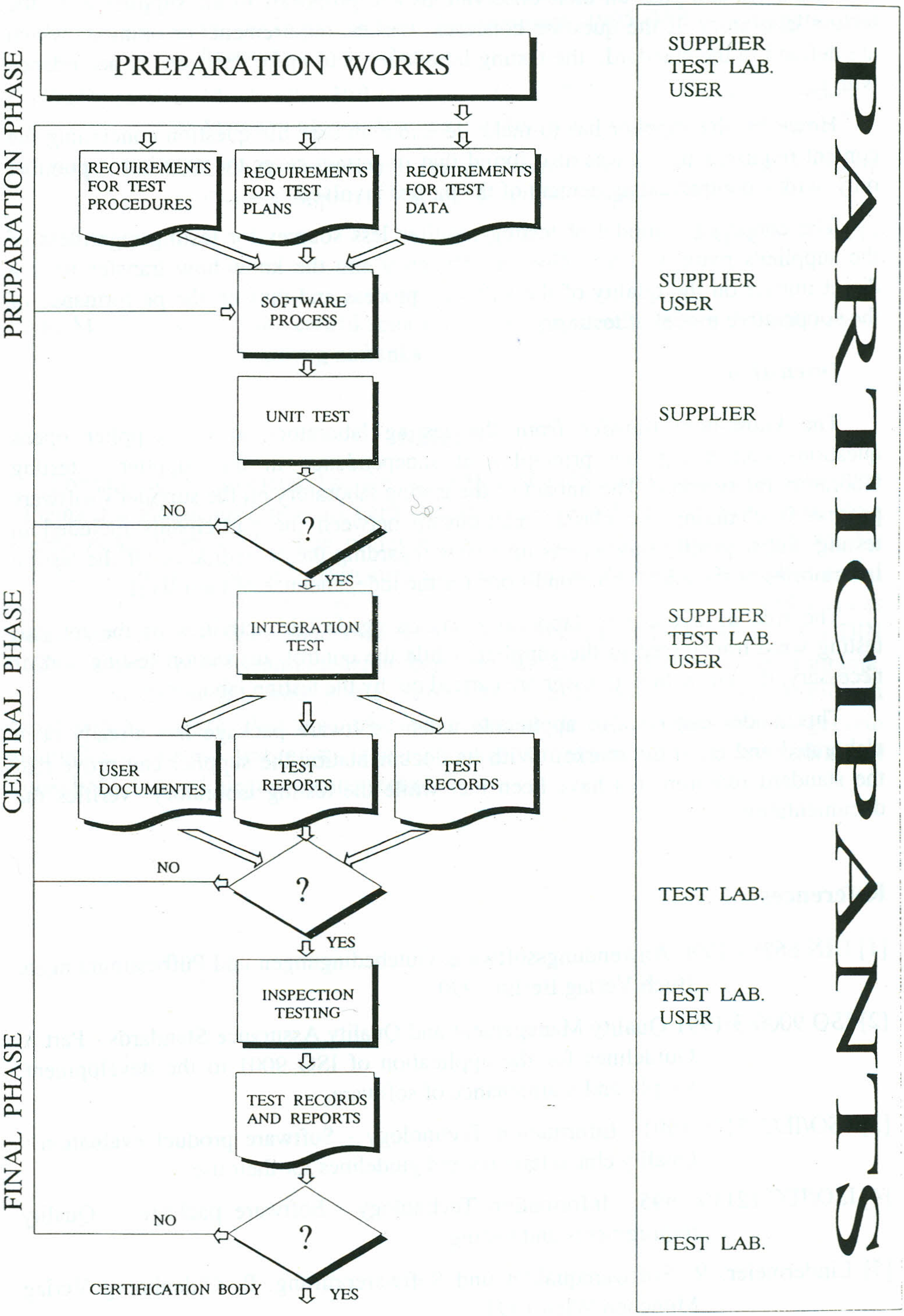

Figure 1: The cooperative model of testing

The users can pass on their observations and proposals to the supplier or to the testing laboratory. If the question concerns wishes, requirements or opinions which are defined in the standard, the testing laboratory is to make decisions in accordance with it.

However, the supplier has to make decisions in case the question concerning the content requirements. It was also found that in certain cases the solution is possible only with a contractual agreement of the parties involved.

The cooperative model of testing requires less sources for testing regardless of the supplier's maturity level. Greater efficiency and the know-how transfer have a direct impact on the quality of the software process and thus on the performance of the cooperative model of testing.

Discussion

The know-how transfer from the testing laboratory to the supplier opens questions concerning the principles of independence in the supplier - testing laboratory relationship. The impact of the testing laboratory on the supplier's software process is changing the classic relationship between the participants included in testing. Subsequently this opens up issues regarding the accreditation of the testing laboratories or the additional conditions for the independence of their work.

The role of the testing laboratory has changed. The activities of the integral testing were transferred to the supplier, while the control, inspection testing and, if necessary, the know-how transfer are carried out by the testing laboratory.

This model can be also applicable when software package has already been elaborated and is on the market. With its documentation the supplier can prove that the standard requirements have been met while the testing laboratory verifies the documentation.

References

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

[2] ISO 9000-3:1991 Quality Management and Quality Assurance Standards - Part 3: Guidelines for the application of ISO 9001 to the development, supply and maintenance of software.

[3] ISO/IEC 9126:1991 Information Technology - Software product evaluation - Quality characteristics and guidelines for their use.

[4] ISO/IEC 12119:1995 Information Technology - Software packages - Quality requirements and testing.

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

[6] Pivka, M. 1995 Software quality system in a small software house. Procceeding of Software Quality Management 1995 Vol.1. Computational Mechanics Publication.

[7] Paulk, et. al: M. Paulk, Bill Curtis, M.B. Chriss&C Webber, Capability Maturity Model, Version 1.1. IEEE Software, July 1993, pp18-27.

[8] Rout, P.T.SPICE: A Framework for Software Process Assesment Software Process. Improvement and Practices. August 1995. Pp. 57-66.

Received: 1996-07-15

Pivka M., Potočan V. Kako certifikacija sofverskog proizvoda može poboljšati proces njegovog nastanka

Sažetak

Poznati modeli procjene, kao što su CMM, BOOTSTRAP, SPICE ili ISO 9000 zanemaruju mogućnost (utjecaj) certifikacije kvalitete softverskog proizvoda. Prva norma za kvalitetu softverskog proizvoda bila je njemačka norma DIN 66285. Na osnovi ove norme ISO je razvio međunarodnu normu za specifikaciju kvalitete i procedure testiranja softverskih proizvoda, ISO/IEC 12119. U ovom radu prikazuju se naša iskustva s klasičnim modelima testiranja baziranim na ISO/IEC 12119 i DIN 66285 i s našim poboljšanim modelom koji smo nazvali kooperativni model testiranja. S ovim modelom možemo postići: poboljšanje efikasnosti testiranja i procesa proizvodnje softvera. Praktična iskustva pokazuju da su se izmijenile uloge svih sudionika u procesu testiranja. Transfer znanja između laboratorija za testiranje i dobavljača ovisi od njegove razine zrelosti.