

Factors of software quality evaluation

- Results of two European surveys -

Teade Punter¹ and Giuseppe Lami²

Abstract

This paper focuses upon restrictions and possibilities of Software Quality Evaluation. Therefore the results of two surveys on evaluation carried out in 1997 are used to analyse five factors of evaluation -objective, subject, moment, actors and techniques- and the cost/benefits of evaluation.

1. Introduction

Software has increasing importance throughout society. Medical devices and flight reservation systems are only two products which fully depend on software. Software quality should therefore be a concern of everyone -see e.g. also [1]. To determine software quality, Software Quality Evaluation is necessary. It is the process of identifying quality targets, selecting product parts, choosing appropriate techniques and tools, performing the measurements and reporting the results.

Two main areas are distinguished within Software Quality Evaluation. First *process evaluation* analyses the software development process and highlights areas for improvement. The assessment can be conducted towards either international standards or a maturity model. Examples are the famous Capability Maturity Model [2] and the Bootstrap approach [3]. *Product evaluation* addresses the fit between customer's needs and the suppliers' promises about the product, often referred to as software product quality. Examples are product certification [4], [5].

Software Quality Evaluation has been identified by many authors as important to achieve software quality -see e.g. [6], [7]. However we found that some practitioners report that - especially product- evaluation is not always appropriate. For example we have noticed that people think that certification of software is nice, but that they are not interested enough to pay for it. Another example is the discussion about ISO 9001, which is sometimes seen as an administrative duty which will not result into better software quality. To conduct appropriate Software Quality Evaluation it is necessary to unveil the possibilities and restrictions of evaluation. This paper addresses these issues by examining basic research questions concerning evaluation:

- why is software evaluated? - is about the objective(s) for evaluation,
- what is evaluated? - is about the subject of evaluation,
- when is evaluation is conducted? - addresses the moment of evaluation

¹ Eindhoven University of Technology (The Netherlands)

² Consorzio Universitario in Ingegneria della Qualità (Pisa, Italy)

- what are the resources necessary to conduct the evaluation? - In this paper resources are split into the people/actors involved in evaluation -their knowledge and abilities- and the techniques to conduct the evaluation.

Without pretending that answering these questions will automatically lead to appropriate evaluation, we think that they will be beneficial because each question addresses input -and with that the possibilities and restrictions- of evaluation. Therefore they are noticed as factors of evaluation.

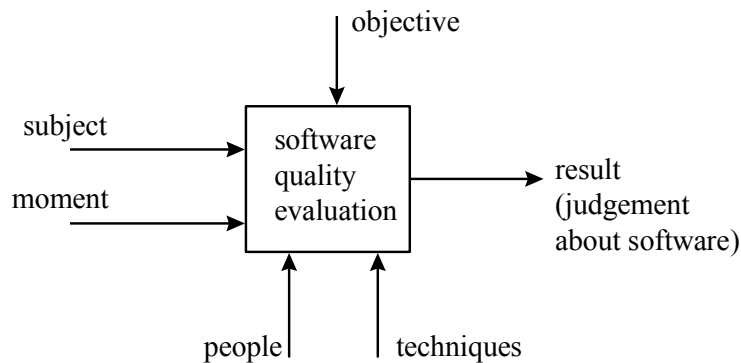


Figure 1 Evaluation factors

When looking at possibilities and restrictions of evaluation it is also necessary to deal with a sixth question: what are cost -e.g. effort- and benefits -e.g. the satisfaction about the result- of the evaluation? The five factors and the latter question are the framework for this paper to analyse Software Quality Evaluation.

2. Research methodology

Two surveys to Software Quality Evaluation were conducted during 1997. The Software Quality Evaluator survey (SQE-survey) [8] has been an analysis on demands and needs in the software quality evaluation field, carried out in the framework of the 'Software Quality Evaluator project' - Leonardo da Vinci programme. This project, co-financed by the EU, has a partnership composed by 14 partners from 3 European countries (Consorzio Universitario in Ingegneria della Qualità (I), GMD (GE), Informatik Apostolidis Hellas (GR), Istituto di Elaborazione dell'Informazione (I), Sistem Form (I), C.U.M. (I), University of Athens (GR), University of Bonn (GE), University of Dresden (GE), HTWM (GE), ADA Italy (I), Delta Barth (GE), Henningsdorfer INO (GE), Synergy (GR)). This analysis has been carried out mainly using a questionnaire, has been oriented to investigate on if an actual place for the professional software quality evaluator exists and which one among the possible professional profiles in the quality evaluation field is the most requested by the job market. The questionnaire was distributed among a selected group of enterprises, which consists of companies involved with project and development of software systems, manufacturing and services. Out of a total of 31 distributed questionnaires, 16 responses were received.

Starting point of the Software Product Evaluation survey (SPE-survey) [9] was the observation that software products are evaluated differently. The survey was started to make an inventory of the factors which will influence the arrangement of those evaluations. The survey consists of a questionnaire for exploratory investigation and was meant to collect the

opinions of people related to quality assessments/ quality assurance. The questionnaire was distributed during four European conferences in 1997: 8th Escom conference, Achieving Software Product Quality conference, 4th Conference on Evaluation of Information Technology and 1st European Software Quality Week. Out of 285 questionnaires distributed, 39 responses were received.

The results of both surveys are used to elaborate on the factors of Software Quality Evaluation. First a general overview of the practice of evaluation is given.

3. Evaluation in practice

Software Quality Evaluation is conducted quite often. 75% of the responding - software engineering, manufacturing and services- enterprises in the SQE-survey execute evaluation. Only considering software enterprises the percentage becomes 100%. The SPE-survey shows that 85% of its respondents are involved in software product evaluation. 33% of these people do have experience for over 10 evaluations. Those figures are supported by [10] which has report that 85% of their responding organisations report that evaluation were executed.

Evaluation is conducted for different kinds of software. Software products that where mentioned during the SPE-survey are: data processing systems, office software, airport resource management and operations system, real time application for telecom, workflow management tool, Case-tools, compilers, safety systems, X-ray-system, sales and strategic marketing system, internet software, infrastructure in public administration, library system, cashing system, warehouse control, shop floor control systems, SAP. So customers and suppliers of a broad range of software products are interested in evaluation.

The SQE-survey shows that software quality evaluation is important (44% of the respondents) or of medium importance (44%) for the strategically framework of the enterprises. When focusing upon the project and development of software systems enterprises, 17% of the respondents think it is important and 83% of them find it is of medium importance.

4. Objective of evaluation (why?)

Respondents were asked to mention the reasons to conduct evaluation (SPE-survey). 67% of them have mentioned determining and/or improving software quality as reason. So software quality should be determined to improve its quality - see also section 6. Other reasons which were often mentioned are: selecting a software package (by 38% of respondents), gaining official acceptance by customer (31%) and reducing uncertainty about the development project (23%). It is remarked that most respondents have mentioned combination of reasons: each respondent has about two reasons. These results are partly supported by the findings of [10]. They report that the most important factors to execute evaluation were to ensure that benefits were achieved and use of evaluation as a learning possibility. Latter is not mentioned by SPE-respondents explicitly. All these reasons reflect the need of confidence between customers and suppliers about the software product and the development process. For example a customer will conduct evaluation to gain confidence in his choice to select the right product. A supplier executes evaluation to know the project risks or to gain acceptance

to his software. To conduct evaluation due to legal regulations was mentioned lesser (by 10%) than the other reasons.

Confidence in software regards the agreement between customer and supplier about the specifications and the extent to which risks are dealt. This is sometimes expressed in terms of evaluation level [6]. Elaborating on the relationship between customer and supplier, SPE-survey shows that evaluations are demanded by the suppliers: management of development project (mentioned by 65% of respondents), purchaser of the product (by 25%) of the product and customers of the product (by 36%). That management of development project is mentioned most often seems to confirm the findings of [11] that 'evaluation is managed and performed, and the evaluation criteria and methods are determined by those who have designed the system'. Regarding the second part of the statement and taking into account that also the evaluation criteria should be set by customers and suppliers to achieve confidence, we have also asked the respondents where criteria of the evaluation are based upon. Contractual criteria of the project (mentioned by 41% of the respondents) and functional requirements (by 30%) and interview with user (20%) are mentioned most often. Especially the contractual criteria and the interview with user refer to the need of customer's involvement, however their actual involvement in it is not confirmed.

5. Subject of evaluation (what?)

To evaluate software, its quality should be specified: requirements to the product are made explicit and often they are expressed as a set of quality characteristics. During the SPE-survey functionality, reliability, usability and maintainability were found the most important characteristics. These four quality characteristics -and also modularity of code/documentation- were also mentioned most often when asking the respondents which attributes were used for evaluation. Internal attributes -like failures observed during running time and number of requirement changes and redundancy in code/documentation were mentioned less often. An exception was size of code. These results points to an interest to evaluate external attributes, which reflect the concept of quality in use [7].

To determine the extent to which quality characteristics are realised by the software product, information should be derived from the product. During the SPE-survey respondents were asked to identify the information source in use.

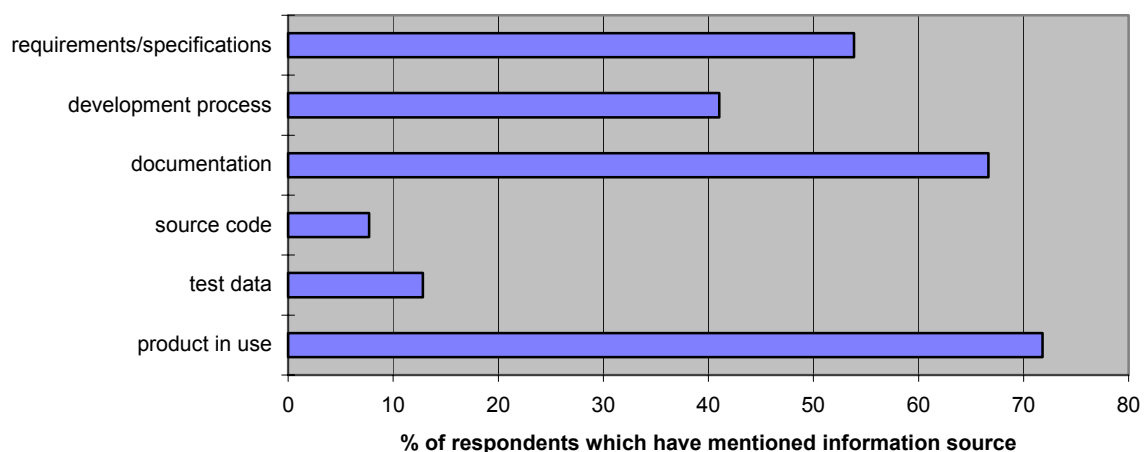


Figure 2 Information sources

The documentation of the software (67%) and information about the software in use (72%) - e.g. opinion of user about the product are mentioned most often as information sources in use. Both are especially available during the life cycle phase operation, see section 6. It is noticed that respondents have mentioned several sources -that why total percentage is more than 100%. Information derived from the specifications (23%) and requirements (28%) and the development process (45%) are important information sources too.

To find why an information source was selected, SPE-respondents were asked for the factor(s) which has influenced the choice of their information sources. Many reasons were given: budget, time to evaluate, availability of (supporting)tools, the agreement between customer and supplier about and preciseness of specifications. All these reasons have been classified into four types of factors.

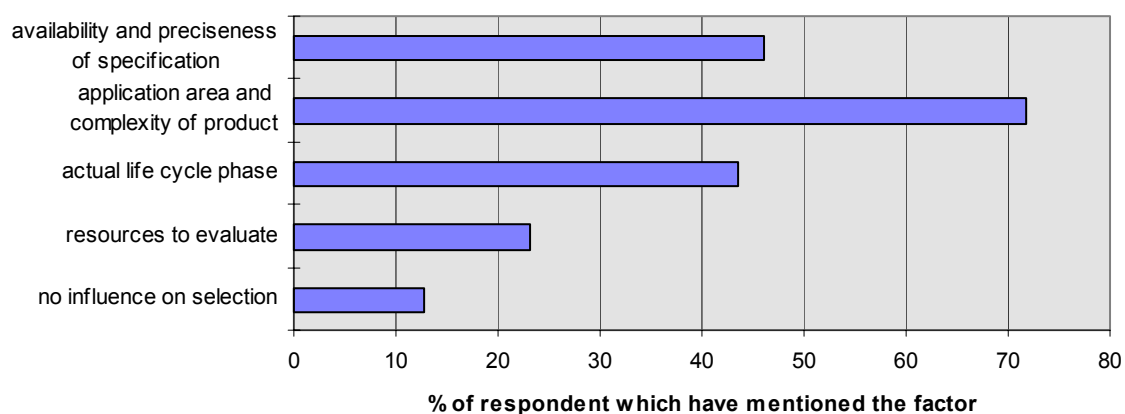


Figure 3 Factors that influence the selection of an information source

We analyse a trend that an information source is selected because of what is known about the application area and the complexity of the product (mentioned by 72% of the respondents). This could be due to the fact that software evaluation is confronted with many different software products that have different application areas and therefore entirely different criteria and information sources to evaluate. The availability and preciseness of the specifications (by 46%) and actual development phase (by 44%) are also important factors. This can be explained by a respondents' remark that explicit requirements specification during the early stages of development, may result into explicit criteria setting -see also section 4. This will influence the number and the kind of information sources taken into account. Requirements can be an information source when explicit requirements setting is conducted at start up of development.

Compared to the other factors, the available resources to evaluate a product -which concerns time and budget to evaluate, availability of tools, competence of evaluator- are of less influence to select an information source (23%). From this we conclude that the way to conduct evaluation -like the specification and the application area and complexity of the product- are more relevant to select necessary information sources -to set up criteria- than constraints like time and budget to go through this information.

6. Moment of evaluation (When?)

Evaluation can be conducted during several life cycle phases of the software product. The SPE survey has made an inventory during which phases evaluation is executed. Therefore five phases were distinguished: 1) acquisition: before start of development or buying the package, 2) development: requirements specification, designing and coding, 3) release (acceptance test): at the end of the development but before operation, 4) operation: use of product and maintenance and 5) new release: at the end of the life cycle of current product, but before new release.

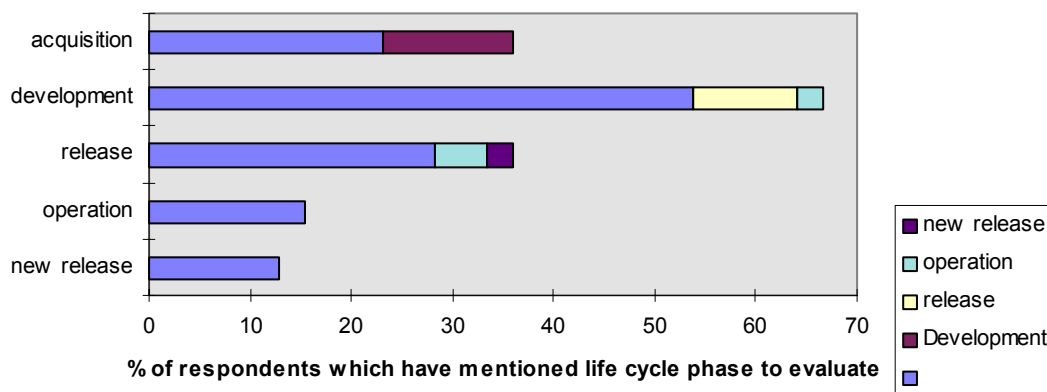


Figure 4 Moment of evaluation

Evaluation is sometimes conducted during several succeeding life-cycle phases. For example: 13% of evaluations during the acquisition phase, was also conducted during the development phase. Evaluation is especially conducted either in parallel with acquisition (mentioned by 23% of respondents) and development (54%) of the product or during its release (28%). Evaluation is not an additional step at the end of the software development, but should result in feedback for development. This result is supported by findings of an earlier survey [13] which found a demand for concurrent assessment: software producers do not want delays caused by the evaluation process and sponsors will know immediately if the product does not satisfy the required level of quality. As a consequence actions might be started as soon as quality deficiencies are detected.

7. People (who?)

Different people are involved in software product evaluation: user of product, customer of product, supplier of product, product engineer, developer of product, product management, evaluator, project management and specialist / consultant (test experts). Persons involved during executing the evaluation are (SPE-survey). The outcomes of the SQE survey show that organisations involved in the survey use especially internal people for the software quality evaluation. In particular, the software companies use internal software quality evaluators both for evaluating external attributes (79%), both for evaluating internal attributes (93%) of produced software. For purchased software products both software organisations (100%) and manufacturers and services (92%) use internal people for the software quality evaluation.

More varied are the outcomes related to the role held into the organisation by the software quality evaluators. People who carry out software quality evaluation for a purchased software product are user (30%), people experienced in software evaluation (22%) or people

competent in software engineering (40%). Software companies involve especially people competent in computer science/engineering (55%) for the evaluation of external attributes of produced software, while users are involved in the 30% of the respondents. The evaluation of the internal attributes of software is carried out principally by the same people involved in some other activities during the software life-cycle (55%), and only for the 39% of respondents by people attending specifically to software quality evaluation exist.

SQE survey gives also information on the requisites that the companies ask the software quality evaluators. These requisites are both related to educational certification, but especially to working experiences. In particular 50% of the organisations consider an educational certification -especially in Degree in Computer Science- and 87% consider an experience in programming for qualifying software quality evaluators. A general consideration can be made about the people carrying out software quality evaluation: professionals attending particularly to software quality evaluation are not yet diffused among companies, the evaluation is usually carried out by people not skilled in it.

8. Techniques

An evaluation technique is a tool and/or set of competences to collect data, to perform interpretation of data during the evaluation. An example is a source code analyser to compute code metrics. Many different evaluation techniques exist and it is hard to classify them in a consistent way. However based upon a distinction of techniques for product evaluation [13] we distinguish: static analysis -during which often metrics are used, dynamic analysis/testing techniques, checklist/audit techniques and modelling techniques. Inspections and reviews of code are added as a separate category, particularly to deal with process evaluation. SPE- and SQE-respondents were asked about which evaluation techniques are in use.

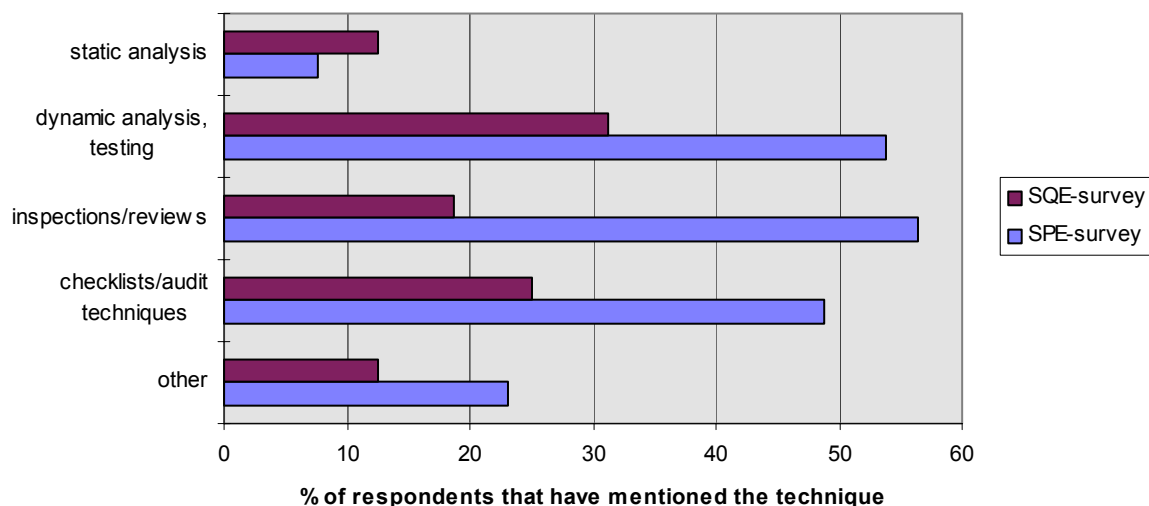


Figure 5 Evaluation techniques in use

Both surveys point out that dynamic analysis/testing and checklists/auditing are the techniques most in use (25% or more). Each SPE-respondent has mentioned two techniques in average. Especially the wide use of checklist/audit techniques points to wide use of qualitative assessment. Reviews/inspections are also commonly used: mentioned by 56% of

SPE-respondents and 19% of SQE-respondents. This denotes to the finding presented in section 6 that a significant number of evaluations is executed during development. The only technique which was -expressed as percentage of total respondents- more mentioned in SQE- (13%) than in SPE-survey (7%) was static analysis. This could be explained by the fact that a large amount of SQE-respondents originate from software industry where the tools to conduct static analysis are more common in use. The category 'other' contains: growth modelling (SPE: 5%, SQE: 0%), interviews/demonstrations (SPE: 13%, SQE: 0%), configuration management (SPE: 0%, SQE: 6%), function point counting/metrics (SPE: 5% , SQE: 6%).

To find why a technique was selected, SPE-respondents were asked for the factor(s) which has influenced the choice of their techniques. Four types of factors has been distinguished - see also section 5.

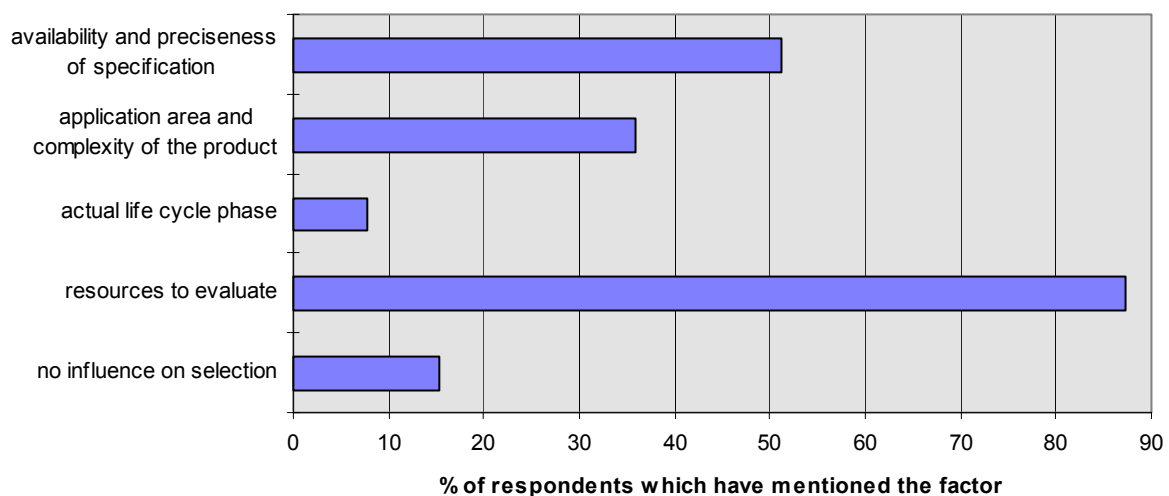


Figure 6 Influencing factors to select an evaluation technique

Deterministic relationships between those factors and a particular technique cannot be given. As the number of respondents per technique raises -see figure 8- the amount of influencing factors was also growing. However figure 9 shows that the most important factor to select a technique are the resources to the evaluation project. For example: limited time and budget to evaluate. This might be a reason to choose a checklist technique instead of building a growth model or execute static analysis.

9. Cost/benefits of evaluation

Previous sections focus on the factors of software quality evaluation, this final section answers the question: 'is the software quality evaluation is carried out effectively?'. Results of the SQE survey about cost/benefits, satisfaction and improvements for evaluation are considered in this section.

The enterprises involved in the SQE survey consider medium/high the effort for software quality evaluation (86% of the respondents). In particular, the estimate of the amount of resources expended for software quality evaluation, as respect as the amount of resources expended for achieving the final product is on the average 18% -it becomes 20% only considering software enterprises. These results come out as quite surprising higher as respect

the values generally recognised, but they are strongly influenced by the particular own software technical area. For example the estimate related to evaluation effort for space software is considerably higher than for administrative software.

The benefits achieved with software quality evaluation involve that software quality evaluation as an activity gives significant benefits on the productivity of the organisation (considered by 54% of the respondents). When enterprises were asked how the effort for software quality evaluation is to be considered in connection with benefits achieved the answer was still: medium/high (81%), it means that the ratio cost/benefits is not thoroughly satisfactory.

SQE survey investigated also directly on the satisfaction for software quality evaluation. Results of this investigation was consistent with results on cost/benefits, in fact 76% of respondents are not fully satisfied of their software quality evaluation. Significant differences emerged between software enterprises and the other enterprises involved, in fact satisfaction is more diffused among the software enterprises (50% of them declare themselves satisfied), but no manufacturing company declares itself fully satisfied of its software quality evaluation.

Improvements needed by the enterprises for software quality evaluation come out particularly interesting with relation to the poor satisfaction demonstrated by the enterprises self involved in the SQE survey. Organisations involved in the SQE survey outlined two main ways in order to achieve improvements in software quality evaluation: educational activities for their internal personnel and increasing the number of their collaborators.

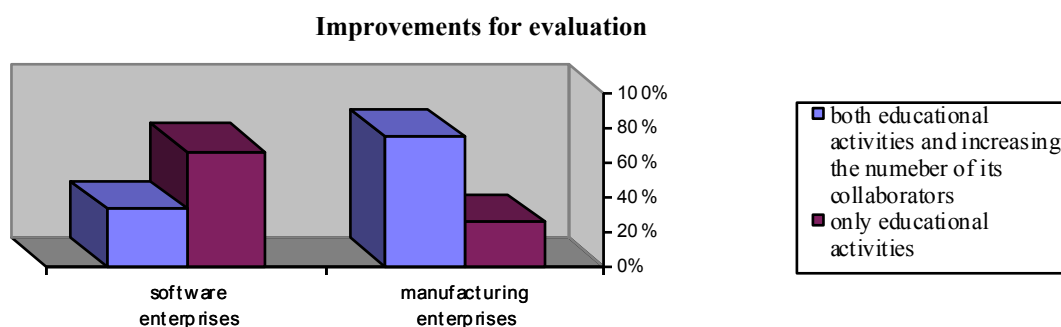


Figure 7 Suggestions for improving evaluation

First outcome is that 81% of all the organisations involved considered carrying out educational activities for their personnel as the way for achieving improvements in the software quality evaluation. A difference between the answers of the software organisations and those of the manufacturing companies still exists. The software enterprises are oriented only in carrying out educational activities for their personnel, but the manufacturing companies are oriented in increasing the number of their collaborators too. The companies that considered increasing the number of their collaborators as a way for solving problems related to software quality evaluation was more oriented forward the direct engagement of personnel not much forward turning to skilled external firms.

10. Conclusions

This paper has presented an overview on Software Quality Evaluation by elaborating on five of its influencing factors and the cost/benefits of evaluation. The dimension of the population of respondents and the fact that two surveys are combined indicate that the results should be treated with a degree of caution. However some trends about the factors of Software Quality Evaluation became clear:

- there is an actual interest in Software Quality Evaluation,
- objective - evaluation is mainly applied to determine and improve software quality. Evaluations are mainly initiated by suppliers of software,
- subject - the scope of evaluations is on external quality of software: quality in use. Information sources most used are information about the product in use and documentation. The most important reason to select an information source is the application area or complexity of the product. We conclude that the way to conduct evaluation -like the specification and the application area and complexity of the product- are more relevant to select necessary information sources -to set up criteria- than constraints like time and budget to go through this information,
- moment - evaluation is especially used pro-active: it is conducted either in parallel with development of the product or before its release. It is not an additional step after development, but it should result in feedback for improving the product,
- people - people involved evaluation are especially part of the organisation's staff. Qualifications requested to people for software quality evaluation are first working experience -especially in programming- and then educational certification -especially degree in Computer Science. The role of the professional software quality evaluator is not diffused among companies, generally the evaluation is carried out by people not skilled in it,
- techniques - dynamic analysis/testing and checklists/audit techniques are the evaluation techniques most often in use: qualitative assessment is applied widely. The fact that also inspections/reviews are applied often, confirms to our findings that evaluation is especially used pro-active. The available resources to evaluate software were mentioned as the most important reason to apply a technique. We conclude that the resources are more relevant to select techniques than the way to conduct evaluation -like specification,
- Cost/benefits - many resources are expended by organisations for software quality evaluation, but full satisfaction is not achieved.

Regarding the dissatisfaction, improvements for Software Quality Evaluation should be made. Suppliers take a lead in evaluation, but to gain customers and users should be involved also. This should be especially done in requirements and criteria setting. The method to specify user requirements which is currently developed by the Space-Ufo project (Esprit II 22290) can be beneficial to this. For the factor people improvements can be made, by achieving more skilled professionalisation in the software quality evaluation. Educational activities for own personnel and increasing the number of skilled personnel -see section 9. In case of the factor technique it is necessary to collect experience data about the resources necessary to apply the techniques. This is to predict budget, throughput and other necessities better for adequate evaluation.

References

1. Wayt Gibbs, W., "Software chronic crisis", in Scientific American, September 1994.
2. Humphrey, W., 'Managing the Software Process', Software Engineering Institute, 1987
3. Kuvaja, P. et al, "Bootstrap: Europe's assessment method", IEEE Software, nr3, pp.93-95, 1993.
4. O'Duffy, M., "Certification of Software for Medical Practitioners", in Veenendaal, E. van and J. McMullan (eds), 'Achieving Software Product Quality', 's Hertogenbosch, Essi Scope, 1997.
5. Pivka, M., "How can software packages certification improve software process", in Proceedings of European Conference on Software Quality, Dublin, 1996.
6. Rae, A., H. Hausen and P. Robert, 'Software Evaluation for Certification', London, McGraw Hill, 1995.
7. ISO FCD 9126, 'Software Quality Characteristics and Metrics', part 1-3, 1998.
8. Lami, G., "Software Quality Evaluation Needs: an empirical study", report of Software Quality Evaluator-project, 1997.
9. Punter, T., "Data about Software Product Evaluation", report Eindhoven University of Technology, 1997.
10. Miller, K. and D. Dunn, "Post-implementation evaluation of information systems/technology: a survey of UK-practice", in Proceedings of 4th European Conference on Evaluation of Information Technology, Delft, 1997.
11. Kumar, K., "Post implementation Evaluation of Computer Based Information Systems: current practices", in Communications of the ACM, 1990.
12. ISO 12207, 'Software Life-cycles', 1996.
13. Bache R. and Bazzana G., 'Software Product Metrics', London, McGraw Hill, 1994.
14. ISO CD 14598, 'Software Product Evaluation', part 1-6, 1996.

Teade Punter is doing a research about Software Product Evaluation since 1996. His interests are the feasibility of third party product evaluation and the means to improve evaluation.

e-mail: h.t.punter@tm.tue.nl

Giuseppe Lami graduated in Computer Science at the University of Pisa in 1994. Since his degree thesis, he concentrated his professional interests on the Software Engineering, and in particular on Software Testing.

e-mail: progetti.qualital@interbusiness.it