

# SOFTWARE QUALITY: WHAT IS REALLY IMPORTANT AND WHO SAYS SO

Maria Sverstiuk\*, June Verner \*\*, Jeffrey Hand \*\*

\* The Wharton School, University of Pennsylvania, 255 South 38<sup>th</sup> Street, Philadelphia, PA 19104,  
Tel 215.573.6842, Fax. 215. 898.2598, {[Sverstiuk@wharton.upenn.edu](mailto:Sverstiuk@wharton.upenn.edu)}

\*\* College Of Information Science And Technology, Drexel University, Philadelphia, PA, 19104  
{[June.Verner@cis.drexel.edu](mailto:June.Verner@cis.drexel.edu), [Jeffrey.Hand@cis.drexel.edu](mailto:Jeffrey.Hand@cis.drexel.edu)}

**Abstract:** More than twenty years ago Boehm, B.W., Brown, J.R., and Lipow, M., [1976] research presented one of the first software quality models. The research identified a hierarchy of software quality attributes. The model became widely accepted for its comprehensive listing of software quality product characteristics. Twenty years later we are reevaluating established hierarchy by putting it in a framework of users priorities. We question what is really important in software quality for different users. To answer this question we conducted a survey of software users ranging from corporate and technical managers to non-technical personnel. We asked them to rank importance of each of the software quality attributes. Results of our study show patterns for software quality preferences for certain groups of users.

**Keywords:** software quality, cognitive model, software quality factors

## Introduction

More than twenty years ago Boehm, Brown, and Lipow, [2], presented one of the first software quality models. Their research provided a detailed definition of software quality and identified a hierarchy of software quality attributes. The model became very widely accepted for its comprehensive listing of quality characteristics. Twenty years later we are reevaluating this quality model within a framework of user priorities in order to identify the most important quality attributes for different user groups. We conducted a survey of software users ranging from corporate and technical managers to non-technical personnel. We asked them to rank the software quality attributes described in the Boehm et al model. The results of our study show very different patterns of software quality attribute preferences and thus different cognitive quality models for different groups of users. A better understanding of users' overall priorities will lead to better communication between the different parties involved in system development and result in the more efficient development of better software systems.

## Related Research

Ideally, every software system should possess the highest measure of quality for each software attribute. However, in reality, everybody involved with the system, from developers to managers and users, has to compromise and focus on the subset of quality factors they believe are the most important. Software quality can be defined from many points of view, depending on the role the person plays in the development process and on the type of system being developed, [1]; [4], [14]. Multiple published studies, based on authors' personal experiences, have identified various relationships between software quality attributes. [9]; [11]; [6], [7], [12].

	Frequency	Percent
Manager	16	18
User	67	74
Developer	7	8
Total	90	100

Table 1. Respondents' roles in respect to the evaluated software

Job Function	Frequency	Percent
Corporate / Technical Manager	51	57
Data Administrator / application developer	1	1
IS/MIS/DP/IT Staff	12	13
Other non-technical	26	29
Total	90	100

Table 2. Respondents' job profile

Job Function \ User Role	Manager	User	Developer
Corporate / Technical Manager	24%	73%	4%
Data administrator / application developer			100%
IS/MIS/DP/IT Staff	33%	50%	17%
Other		92%	8%

Table 3. Distribution of the user roles by job function

The majority of respondents identified themselves as users of the software that they were assessed; 24% of our respondents were both managers by title, and by their software role; 33% of the IS staff were managers of the software they assessed and 17% of them participated in the development of the software they assessed. This overlap between job function and software role allowed us to identify patterns of perceptions of software quality within quite different professional and user groups. In the next section we present detailed results of our study.

## Results

The main purpose of our study is to explore differences in cognitive mappings and prioritization of software quality attributes within different occupational groups. Our sample is divided into groups in two ways: (1) by the respondent's role with regard to the evaluated software (either developer, user, or manager); and (2) by job function (i.e., technical staff, non-technical staff (other), or corporate manager). With these groupings we identify how the perceptions of software quality attributes change with changing software roles. Viewing mean frequency distributions for both the user groups and the job function groups, we identify seven software quality attributes that are consistently ranked the most important: Correctness, Accuracy, Usability, Robustness, Communicativeness, Integrity, and Maintainability. Figure 1 shows the mean distribution of these seven attributes rankings by the three software role groups.

Figure 1 shows the similarities in the different groups' perceptions: Correctness and Accuracy were consistently ranked the most important across all three groups. Respondent perceptions of Communicativeness were also very similar across software roles. However, while Maintainability was scored as important by developers and less important for managers and developers, it scored very low on the users' priority scale. Differences were also observed for Robustness; this attribute was very important for managers and developers, but much less important for users. Usability is more important to users and developers than it is to managers. Significant differences in the scores provided by user role groups were observed for the following attributes (note that we list the attribute, the significance and the group that scored the attribute highest): Maintainability (0.032, Developers), Accountability (0.002 Managers), Efficiency (0.006, Developers), Flexibility (0.000, Managers and Developers), testability (0.003, Developers).

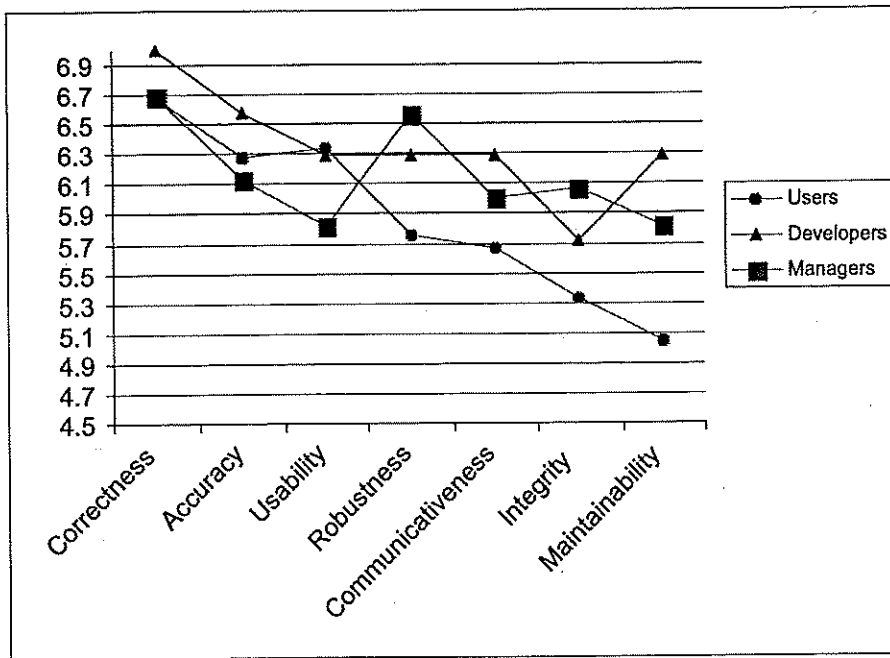


Figure 1. Comparison of software quality attributes mean frequency distribution between different role groups of respondents.

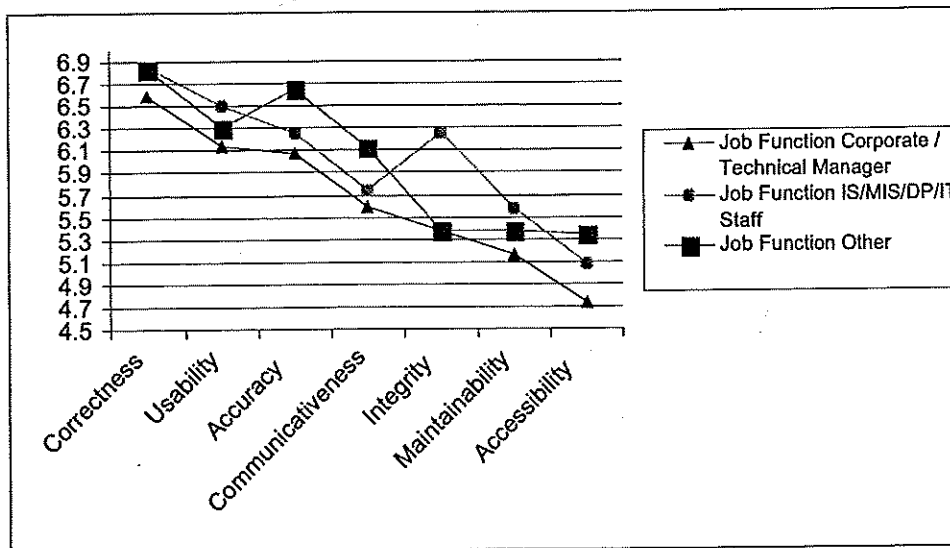


Figure 2 differences in the priorities for software quality as identified by different professional groups.

Figure 2. displays the similarities and differences of the different professional group's perceptions of the software quality attributes. The overall picture is more consistent across different job functions. For the shown attributes, with the exception of integrity and accuracy, scores were very close for managers, technical staff, and non-technical personnel. The only attribute showing a significant difference by job function was Reusability which technical/corporate managers and 'other non-technical personnel' rated significantly higher than the other two occupational groups

The above graphs and analysis shows that it is the role that the respondent takes with regard to the software that colors his or her perceptions of software quality attributes and their importance rather than the professional group that the respondent belongs to.

## Conclusions

Our study identifies the mental or cognitive models of software quality held by different professional groups and how this is affected by the role that the professional takes with regard to software that is being evaluated. Though there are many similarities in quality priorities, there are also differences in perceptions across different roles that the professional may take with regard to the software. Although respondents may discuss the same quality factors they have quite a different view of the importance of some of those factors. While some software quality attributes are important to all users regardless of their job function, the results of the study reveal patterns in software quality priorities for each of the different user profiles.


Differences in perceptions of what constitutes good software require that we identify priorities for all software quality attributes at the early stages of the software development life cycle. While we are aware that our sample size is too small to draw general conclusions, and that differences in perceptions might be caused by our use of a software quality model developed in the mid seventies, we are planning to address this issue in our next study where we will compare perceptions of software quality with and without attribute definitions. We will also include attributes from additional software quality models.

Our current study is an initial step in the development of a tool for early application in the software development lifecycle. This will help to establish better communication between the different professional groups involved in the development process. Making explicit, software quality priorities early will improve our chances of software development success.

## References

- [1] Arthur, L. J. Measuring Programmer Quality, John Wiley and Sons New York 1985
- [2] Boehm, B.W., Brown, J.R., and Lipow, M. (Eds.), (1976), "Quantitative Evaluation of Software Quality", Proceedings of the Second International Conference of Software Engineering, pp. 592-605.
- [3] Cerpa N. and Verner J.M., "Practitioners' Views of Software Quality", The International Journal of Computing and Engineering Management, Vol. 4, No. 3, September-December 1996, pp. 1-15.
- [4] Deutsch, M. S and Willis, R. R. Software Quality Engineering Prentice-Hall Englewood Cliffs NJ (1988).
- [5] Gentleman, W.M., "If software quality is a perception, how do we measure it?", The Quality of Numerical Software: Assessment and Enhancement, Ronald Boisvert, ed., Proceedings of IFIP WG2.5 Working Conference 7, Oxford, UK, 7-12 July 1996, Chapman & Hall, London, p.32.
- [6] Gillies, A. Software Quality Theory and Management, 2<sup>nd</sup> Ed., London: International Thomson Computer Press, 1997.

- [7] Glass, R.L. "Building Quality Software", Prentice Hall, 1992
- [8] Kusters, R.J.; van Solingen, R.; Trienekens, J.J.M., "User-perceptions of embedded software quality", Eighth IEEE International Workshop on Software Technology and Engineering Practice incorporating Computer Aided Software Engineering, p.184-97, 1997
- [9] McConnell, Steve. Code Complete: A Practical Handbook of Software Construction. Redmond, WA: Microsoft Press, 1993.
- [10] Nance, K.L., Strohmaier, M., " End user perception and software quality assessment", Journal of International Information Management vol.6, no.1, p.1-8, 1997.
- [11] Perry W.E. "Quality Assurance for Information Systems: Methods, Tools and Techniques", QED Technical Publishing Group, 1991.
- [12] Shumskas, A. F. "Software Risk Mitigation" in Schulmeyer, G. Gordon and James I. McManus, ed. Total Quality Management for Software. 1992: NY, Van Nostrand Reinhold. pp. 190-220.
- [13] Verner, J.M., Moores, T.T. and Barrett, R., "Software Quality: Perceptions and Practices in Hong Kong", in Achieving Quality in Software (eds) S. Bologna and G. Bucci Chapman and Hall 1996, pp. 77-88.
- [14] Wallmuller, E., Software Quality Assurance: A Practical Approach, Prentice-Hall Englewood Cliffs NJ 1994.
- [15] Wilson, D. N. and Hall, Tracy, "Perceptions of Software Quality: A Pilot Study". Software Quality Journal, 7 pp. 67-75, 1998



**Nimestic**  
10 001 2000

**Congrès scientifique**

**International conference**

**11, 12 et 13 Septembre 2000**  
**11- 13 September 2000**

**Nîmes - France**

**Ingénierie des systèmes  
et NTIC**

**Systems engineering  
and information  
& communication  
technology**

[www.nimestic.com](http://www.nimestic.com)

Tél. (33) (0)4 66 38 70 02