



## TOWARDS IDENTIFYING ASPECTS OF SOFTWARE QUALITY

Dr. Latika Kharb<sup>1\*</sup>

<sup>1\*</sup>Jagan Institute of Management Studies (JIMS), Rohini, Delhi, India.

Correspondence Author: [latika.kharb@jimsindia.org](mailto:latika.kharb@jimsindia.org)

**Keywords:** Identify quality, Software Quality Characteristics

### Abstract

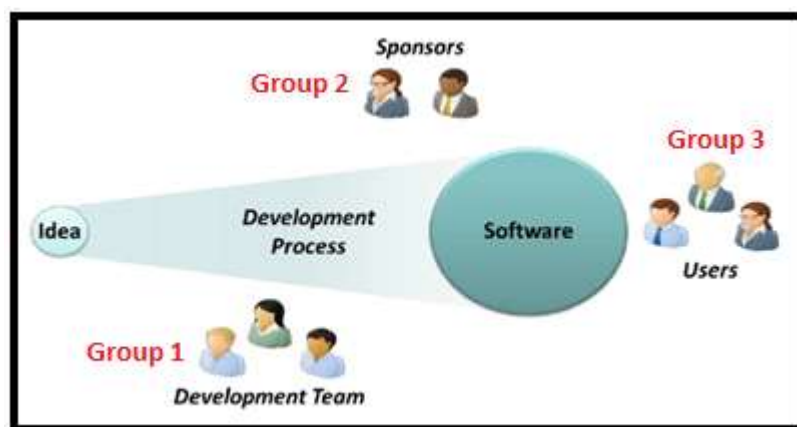
Software projects are notorious for going past their deadline, going over budget, or both and the problem lies in the estimation of the amount of effort required for the development of a project. Reliability in software is produced by controlling its quality within budget. In early days of computing, software costs constituted a small percentage of overall computer- based system costs but today; software is the most expensive element of virtually all computer based systems. In this paper, we have discussed about the concept of software quality, the various group of people involved and their roles and various aspects of software quality. We have tried to identify the fact that understanding the reality and importance of software quality is the need of hour as the low-quality software just isn't acceptable in present and/or future time span.

### Introduction

Software projects are notorious for going past their deadline, going over budget, or both and the problem lies in the estimation of the amount of effort required for the development of a project [1]. Today, anyone can conclude that half of the entire world runs on software: every business depends on it, every mobile phone uses it, and even every new car relies on code and so on. Without software, modern civilization would fall apart. By understanding this reality, the quality of that software really matters: as it's so widely used and so important, the low-quality software just isn't acceptable. Software quality is the degree of conformance to explicit or implicit requirements and expectations [2]. We can elaborate this as:

1. **Explicit:** clearly defined and documented.
2. **Implicit:** not clearly defined and documented but indirectly suggested.
3. **Requirements:** business/product/software requirements.
4. **Expectations:** mainly end-user expectations.

Reliability in software is produced by controlling its quality within budget. In early days of computing, software costs constituted a small percentage of overall computer- based system costs but today; software is the most expensive element of virtually all computer based systems [3]. One useful way to think about the topic is to divide software quality into three aspects: Functional quality, Structural quality, and Process quality.



*Figure 1: Three groups of people involved in Software Quality*

A development process converts an idea into usable software. In thinking about software quality, it's useful to focus on three groups of people who care about its value: software users, development team and sponsors of project.

1. **Software users:** First group of people is software users; who apply this software to some problem. Users care primarily about functional quality, since that's what they see. They will also care about some aspects of process quality, such as the delivery



date of the final software. But, users typically don't care at all about structural quality, even though its absence might well impact them over the usage of software for lifetime.

2. **Development team:** Second group of people is the development team that creates the software; they certainly does care about structural quality, as they're the only group of people who will be affected by the problems caused by low quality. They also care about functional quality, although perhaps a bit less than users do—cutting features that users want can make life easier for developers. Development teams also care about process quality, in some percent because it provides many of the metrics by which they're measured.
3. **Sponsors :** The third group of people is sponsors that care about all aspects of quality: functional quality, structural quality and process quality. Sponsors strive for business value, so they are interested in taking a broad view of software quality. So, they must also consider the connection between quality and risk. The sponsors of the project are the people paying for the software's creation.

Unsurprisingly, everybody involved in a software project cares most about the aspects of quality that directly impact them. All three of these groups care about software quality. The function of software quality that checks that the project follows its standards processes, and procedures, and that the project produces the required internal and external (deliverable) products [4].

### Aspects of software quality

The goal of software quality assurance is to assure that the quality of the new software system version does not fall below that of the previous version [5]. In this section, we'll discuss about the three aspects of software quality namely, *functional quality*, *structural quality*, and *process quality*.

**Functional Quality:** Functional quality means that the software correctly performs the tasks it's intended to do for its users. Functional quality includes numerous attributes like:

1. **Meeting the specified requirements.** It is one of the essential attribute of functional quality: whether it's related to project's sponsors or software's intended users. And since requirements commonly change throughout the development process for achieving this goal, it requires the development team to understand and implement the correct requirements in depth.
2. **Creating software that has few defects.** It includes bugs that reduce the software's reliability, compromise its security, and/or limit its functionality.
3. **Good enough performance.** It's one of the most important attribute of functional quality from a user's point of view, there's no user that demands for a good but slow application.
4. **Ease of learning and ease of use.** It's another important attribute because for the users, the software's effective and easy to use user interface is everything for a good performance of workflow.

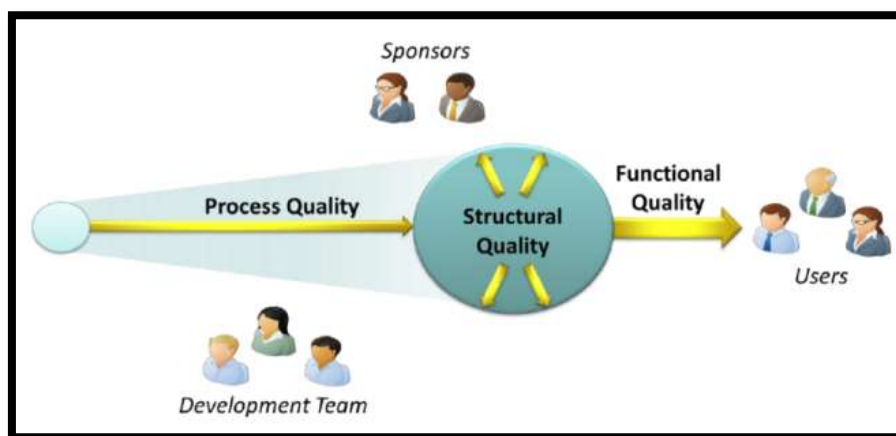


Figure 2: Aspects of Software Quality



**Structural Quality:** The second aspect of software quality, structural quality, means that the code itself is well structured. Unlike functional quality, structural quality is hard to test for (although there are tools to help measure it). The attributes of this type of quality include:

1. **Code testability.** It describes if the code is well organized to make testing easy?
2. **Code maintainability.** How easy is it to add / change/ modify the existing code without introducing bugs?
3. **Code understandability.** It has a great impact on how quickly new developers can begin working with an existing code base: so check for code readability, code complexity, code efficiency etc.
4. **Code security.** How much the software is secure against buffer overruns/ SQL injections etc.

**Process Quality:** The third aspect, process quality, is also critically important. The quality of the development process significantly affects the value received by users, development teams, and sponsors, and so all three groups have a stake in improving this aspect of software quality. The most obvious attributes of process quality include these:

1. **Meeting delivery dates.** Was the software delivered on time?
2. **Meeting budgets.** Was the software delivered for the expected amount of money?
3. **A repeatable development process that reliably delivers quality software.** If a process has the first two attributes—software delivered on time and on budget—true process quality means being consistent from one project to the next.

You're never completely sure that the project plan is right until you've delivered a high-quality product on time and within budget. However, a project manager does it right when he encourages software people to work together as an effective team, focusing their attention on customer needs and product quality [6].

In general, each development project weighs the interests of all three groups—and all three aspects of quality—against one another. Software quality is built from the ground up, with design and development methodologies, and with a special focus on testability, coverage, and flexibility [7].

## Conclusion

During analysis over software quality, the three aspects and group of people involved in it, we come to the conclusion that there's no single best way to view software quality—different perspectives emphasize different things. The three-part breakdown described in the paper provides one useful way to think about this area as research. We discussed how by providing an equal emphasis on functional quality, structural quality, and process quality, it helps broaden our view to include the things that matter to all three stakeholders: users, development teams, and sponsors.

## References

1. Kharb L, Evaluation of Total Cost of Testing Through Proposed Cost Estimation Metrics; Proceedings of International Conference on Trendz in Information Sciences and Computing (TISC2007) Sathyabama University, Chennai, India: 12-14 December, 2007; Volume II, Pg: 677-680.
2. <http://softwaretestingfundamentals.com/software-quality/>
3. Kharb L, Proposed C.E.M (Cost Estimation Metrics): Estimation of Cost of Quality in Software Testing, International Journal of Computer Science and Telecommunications, Volume 6, Issue 2, February 2015, Pg 10-14.
4. <http://www.sqa.net/softwarequalitycontrol.html>
5. Daniel Galin: Software Quality Assurance, Pearson Education.
6. Pressman, R. S. (2000) Software Engineering – A Practitioner's Approach, McGraw-Hill International, London.
7. <http://www.ocoudert.com/blog/2011/04/09/what-is-software-quality/>