

**REQUIREMENTS ENGINEERING
FOR BUSINESS STAKEHOLDERS:
APPLYING QUALITY VIEWS FRAMEWORK TO SOFTWARE**

M. HAIGH

We conducted a survey of more than 300 business stakeholders, asking them to specify their views on software quality requirements within established quality framework. The results showed business role-related differences in specific areas. The paper also considers the implications of these results and their relevance to software requirements analysis.

INTRODUCTION

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]. Garvin [4] generalized these differences in perceptions in a quality framework applicable to design and manufacturing processes of all kinds. He identified five main views of quality: transcendental, product, user, manufacturing, and value-based views. Managers, technical personnel and customers all might differ in their views of what contributes to quality of the software. While, some organizations may have no actual quality definition, in other organizations the view of software quality depends on the occupation of the person establishing the definition or the maturity of the software development process [5–7].

Differences in perceptions of software quality and their impact on the software product development might imply a need for more careful and explicit attention to be paid to the setting of agreed levels for each quality attribute. For example, managers might need to understand what aspects of software quality are most important to users to ensure that system developers implement the most important features when resources are constrained, or when the quality attributes are in direct opposition to each other. On the other hand, the concerns of developers may reflect technical characteristics of the system that — in a manner not fully apparent to managers or users — underpin the delivery of attributes of more explicit concern to all stakeholders.

This study investigates whether managers, developers, and users subscribe to different perspectives on software quality within Garvin's model. It is the first to date to apply Garvin's framework of quality to software quality.

BACKGROUND

Garvin (1984) suggested that the interpretation of quality depends on who is defining it. His definition includes five overarching dimensions of quality. Even though Garvin's framework was developed with a general quality concept in mind, it can be usefully applied to the case of software quality. The dimensions of the Garvin's framework are as follows:

- The *transcendental view*: mostly relates to the elusiveness of the quality concept. Within this view quality is defined as «innate excellence». It is assumed that intuitively everybody realizes a quality product when they see it, but that quality cannot be defined precisely.

- The *product view* sees quality connected to essential characteristics of the product. Measuring systems internal properties offers an objective and context-independent view on quality. This view leads to the quantifiable view of quality. It implies that quality attributes can be unambiguously enumerated and hierarchically organized. Many models of software quality have been derived based on this product view of quality. [8] point out that more research is needed to confirm a positive correlation between these «internal» and the «external» utility of the product in the social setting for which it is being designed.

- The *user view* assesses product quality in a task context. This view defines quality in terms of fitness for purpose. Quality is shown by how well the software meets the needs and preferences of a specific user during its actual use.

- The *manufacturing view* evaluates quality as a measure of the effectiveness and reliability of the process by which the software is produced. This view of quality results in a process assessment that is independent of the product itself and instead examines whether the product was developed in the most cost efficient way. This view of quality implies that there is direct correlation between the development process and its outcome: the premise is that a better development process will lead to a better outcome.

- The *value-based view* assesses quality in terms of its importance to a customer. In other words quality depends on how much customer is willing to pay for a certain quality attribute. The value-based view is defined through relationships or tradeoffs between various quality attributes. The value-based view is different from the user view of quality because it focuses on tradeoffs between cost and quality, not necessarily on user needs.

Even though Garvin's framework has never been directly applied to software quality, it accommodates and illuminates many of the software quality models developed over the last 20 years.

METHOD

For this study an online survey of 315 software stakeholders was conducted. The survey made available using a web interface connected to a database. The URL was distributed via email to the Executive MBA students and alumni at one of the most highly ranked business schools in the United States. Distribution of the survey to this sample facilitated reaching a homogeneous group of people with the same education, yet representing managers, users, and technical personnel from all sectors of the U.S. economy. Various aspects of this elaborate study had been examined elsewhere [9–11].

Respondents used a wide variety of different software packages. The survey therefore asked each respondent to select the piece of software most important to them in carrying out their work responsibilities and answer questions with respect to this piece of software. This gives more meaningful results than simply asking the respondent about his or her attitudes to software in general.

Stakeholder role was defined with respect to the specific piece of software chosen for evaluation. We used two axes on which to divide our respondents into four distinct software stakeholder roles. There is an axis of users versus developers: stakeholders who are involved in managing or performing the software development process and those who are not directly involved in these tasks. There is also an axis of managerial versus non-managerial responsibilities (with regard to the specific piece of software evaluated).

The focus of this study is to find out whether members of the four different stakeholder groups exhibit widespread and systematic divergences regarding software quality. Thus the research question of the study is as follows: *Are there systematic differences between different software stakeholder groups in their endorsement of different views on quality as defined by Garvin's framework?*

The null hypothesis of the study can be expressed as follows:

H0: *There is no significant difference in software quality views between different software stakeholder groups.*

The corresponding alternative hypothesis is thus:

H1: *There is a significant difference in software quality views between different software stakeholder groups.*

DEMOGRAPHIC DATA

The survey included questions covering stakeholder's job function, their relationship to software product most important for their job function and their views on software quality.

Each respondent identified him- or herself as either a user or developer of the software concerned, and as either a manager (managing its users or developers) or non-manager (personally using or developing the software concerned). Combining these two variables thus divided respondents into four groups, which we refer to here as stakeholder roles: User, Manager of Users, Developer, and Manager of Developers. Thirty one percent of the respondents were responsible for development of the software concerned: 16.2 % were managing its development, while a further 14.6 % were personally performing development tasks. The remaining 69 % of the respondents were not associated with the development of the software evaluated, and are therefore treated here as users. Fifty percent personally used the software they evaluated and 18.7 % identified themselves as managers of the users of the software they evaluated (35% of the respondents fell into one or other of the management roles). Most of the respondents (60 %) came from two sectors: (1) IT and Telecommunications, and (2) non-IT services. Overall, however, seven major industry categories were represented.

Table 1 shows the distribution of stakeholder roles by industry. Responses associated with developers and developer managers mainly came from IT and Telecommunication industries: 43 % and 44 % respectively. The service-non-computer industry was the most represented for respondents not associated with

software development: 39 % of software users and 32 % of user managers were from this industry. While each stakeholder role was found across the full range of industries, there is clearly some covariance between industry and role — some of which may reflect the nature of each industry and some of which may be due to random variation in the sample.

Table 1. Stakeholder roles by industry

Industry (column %)	Dvlp. <i>n</i> = 46	Mgr.Dvlp. <i>n</i> = 52	User <i>n</i> = 155	Mgr.Use <i>m</i> = 59
IT and Telecomm. <i>n</i> = 92	43.4	44.2	21.3	25.4
Government <i>n</i> = 16	10.9	1.9	3.4	6.8
Healthcare <i>n</i> = 32	6.5	7.7	12.3	10.2
Manufacturing <i>n</i> = 55	13.1	13.5	18.7	22
Military <i>n</i> = 5	2.2	3.9	0.7	1.7
Academic and Research <i>n</i> = 15	6.5	11.5	3.2	1.7
Service-Non-Computer <i>n</i> = 100	17.4	17.3	40	32.2

Respondents evaluated a variety of software packages. These packages were categorized across two axes:

- software application area: business administration, manufacturing or production, scientific/research activities, creativity-related software (e.g., games, art/graphics, music, etc.), and other;
- software type: off-the-shelf-software; off-the-shelf-software customized for respondent's company use, in-house developed software for sale, in-house developed software for the use within respondent's organization, and «other», software did not fit into any of the previous categories.

Forty seven percent of the respondents evaluated business administration software, making this by far the most represented category of software in the survey. Thirty two percent of the software evaluated was categorized as «other» — meaning that the respondent did not believe it to fit into any of the pre-defined application area types. Scientific and manufacturing software were other two most popular application areas (9.5 % and 8.9 % respectively) (Table 2).

Table 2. Software application area chosen for evaluation by stakeholder role

Appl. Area (Column %)	Dvlp. <i>n</i> = 46	Mgr. Dvlp. <i>n</i> = 52	User <i>n</i> = 155	Mgr. User <i>n</i> = 59
Business Admin. <i>n</i> = 147	37.8	30.6	59.7	37.9
Creativity <i>n</i> = 4	0.0	0.0	2.0	1.7
Manufact. <i>n</i> = 28	8.9	24.5	2.0	15.5
Other <i>n</i> = 100	44.4	24.5	28.6	37.9
Scientific <i>n</i> = 30	8.9	20.4	7.8	6.9

Table 2 shows the software application areas evaluated by respondents in different stakeholder groups. Data in this table reflects missing data and rounding errors.

Table 3 shows that sixty two percent of users primarily used off-the-shelf software for their business responsibilities. Developers and developer managers

were involved with in-house software developed for sale, off-the-shelf customized software, and in-house developed software for internal use only. Business stakeholders along the managerial axis commonly used off-the-shelf customized software and in-house software developed for the use within their own organization.

Table 3. Software type chosen for evaluation by stakeholder role

Software Type (Column %)	Dvlp. n = 46	Mgr. Dvlp. n = 52	User n=155	Mgr. User n=59
Off-the-shelf-software	15.2	5.8	62.6	20.3
Off-the-Shelf-Customized	17.4	25.0	19.4	45.8
In-house developed to sell	39.1	32.7	7.1	8.5
In-house developed for the use within own organization	23.9	28.9	9.0	20.3
Other	4.4	7.7	1.9	5.1
Total	100	100	100	100.0

Table 4. Average satisfaction with evaluated software by stakeholder groups

Stakeholder Role	Satisfaction Avg
Developer	3.78
Manager of Developers	3.88
User	3.95
Manager of users	3.91

Respondents were reasonably happy with the software under consideration: 78.2 % measured their satisfaction with the software as '4' on a 7-point scale (Table 4).

In the next section we present the results of our analysis of the stakeholders' quality views regarding software used for their jobs.

DATA ANALYSIS

One question in the survey presented respondents with five statements on software quality, each designed to correspond with one of Garvin's perspectives on quality. Respondents were required to choose only one view. Results are below, together with the five statements themselves.

Table 5. Software quality views choices for all respondents

Statement on Software Quality	Garvin View-point	Number Choosing	Percentage Choosing, %
«Software quality is shown by how well the software meets the needs and preferences of a specific user during actual use»	User View	221	70
«Software quality is always a tradeoff between acceptable levels of excellence and cost»	Value View	46	15
«Software quality is best assessed by looking at the process of the software production process»	Manufacturing View	30	10
«Software quality can be recognized, but not formally defined»	Transcendental View	10	3
«Software quality is best assessed by looking at the internal qualities of the program code and comparing them to standard measures»	Product View	7	2

A frequency distribution of the software quality views by stakeholder groups showed consistent views across all groups (Table 5). Fig. 1 and Fig. 2 show distribution of software quality views by stakeholder groups.

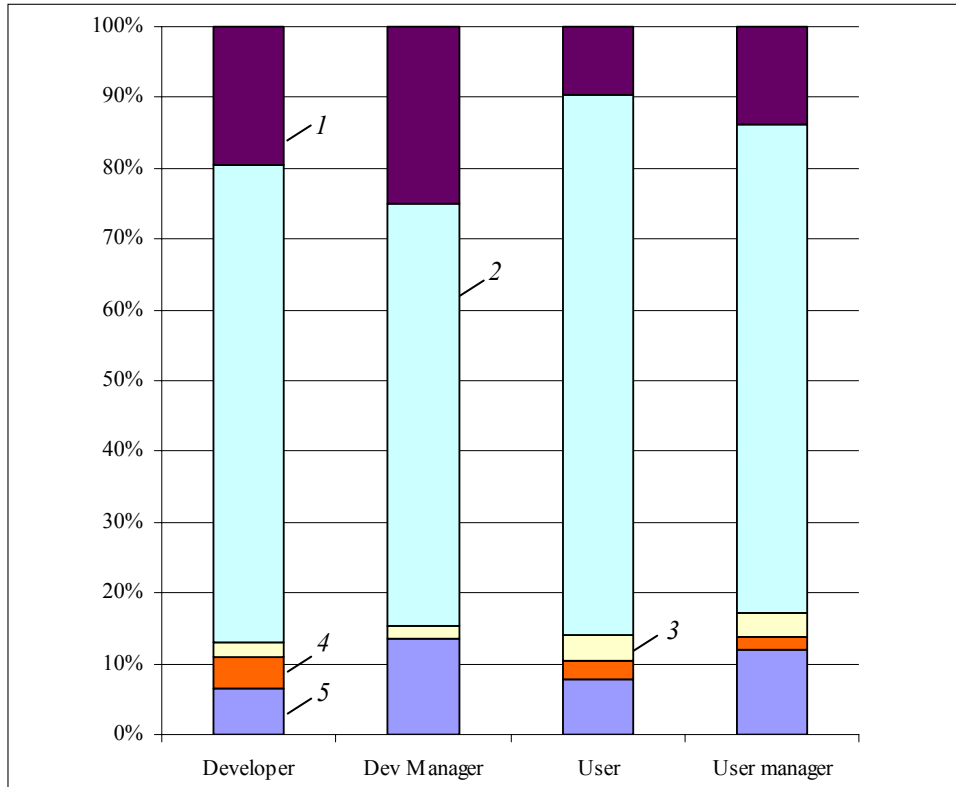


Fig. 1. Software quality views choices by stakeholder role: 1 — Value view, 2 — User view, 3 — Transcendental view, 4 — Product view, 5 — Manufacturing view

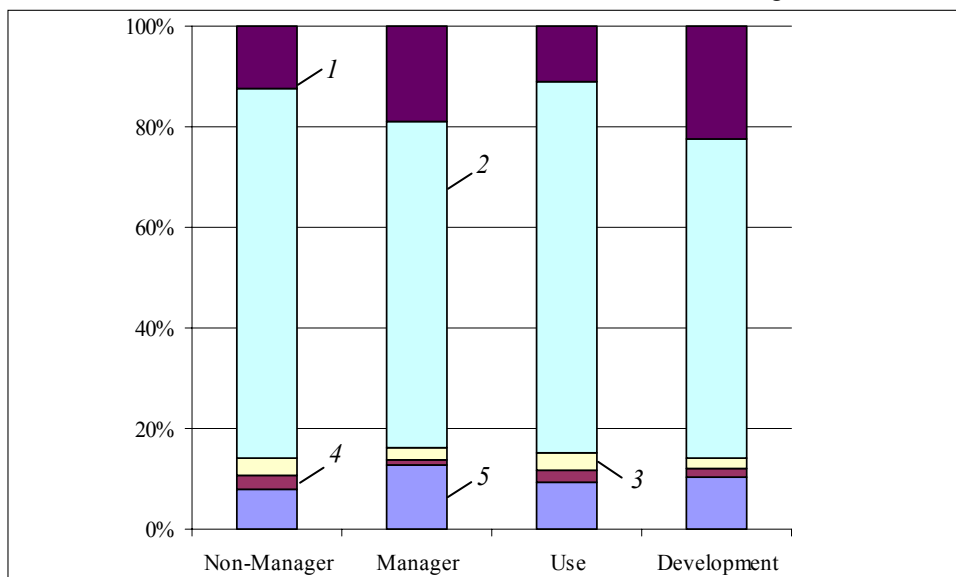


Fig. 2. Software quality views choices by aggregate stakeholder groups: 1 — Value view, 2 — User view, 3 — Transcendental view, 4 — Product view, 5 — Manufacturing view

Users accounted for 72 % of the aggregate use group (together with user managers) and 77 % of the aggregate non-management group (together with developers). Crosstabulation analysis showed no statistically significant differences between stakeholder groups in their view of software quality ($p > 0.29$). However the value view of quality is noticeably more popular with managers, particularly managers responsible for the development process, than with users.

The user view was the most frequently adopted view among all groups, with the value view the second most agreed view for all stakeholder groups; and manufacturing view was the third most commonly endorsed. Clearly, most stakeholders believe that software quality can be defined, but cannot be identified by the application of formal measurement of the code itself (unpopularity of the product view manifested this).

Respondents who took the product view of quality showed the most pronounced differences in quality attribute importance: integrity and interoperability were ranked much higher than among respondents with other views. But because this view and the transcendental view were infrequently reported this result is less significant. However, the differences between respondents taking the manufacturing view and those taking the user and value views are quite striking.

CONCLUSIONS

The user view of software quality (software quality is shown by how well the software meets the needs and preferences of a specific user during actual use) was the most frequently chosen response. The value view (software quality is always a tradeoff between acceptable levels of excellence and cost) was the second most popular. The manufacturing view (software quality is best assessed by looking at the process of the software production process) was picked by 10 % of respondents. The product and transcendental views were the least popular. Only 2 % of the respondents agreed with the product view (software quality is best assessed by looking at the internal qualities of the program code and comparing them to standard measures). Only 3 % of the respondents chose the transcendental view (software quality can be recognized, but not formally defined).

Although the user view was popular with all stakeholder groups, it was most popular among users. The user view defines software quality by how well the software meets the needs and preferences of a specific user during actual use. Users preferred to adopt the view of quality that focuses on them: software is good when it satisfies their needs. However, its appeal was clearly more general and may be attributed to a general sense that the quality of software is hard to define or measure more formally. In addition, those involved in developing software may be consciously attempting to see matters from the viewpoint of their users and customers.

The value view defines software quality as a tradeoff between acceptable levels of excellence and cost. The value view was more popular with development managers than with any other group — perhaps, because they have responsibility for making tradeoffs and satisfying users within cost constraints.

Both of the managerial groups (development managers and user managers) chose the manufacturing view more often than did either of the non-managerial groups (users or developers). The manufacturing view presumes that software

quality is best assessed by looking at the process of the software production process. This phenomenon could be explained by the fact that these respondents may have more faith in the management process by the nature of their managerial responsibilities toward the software. They are also likely to feel that it is through the establishment of a sound process that their personal contribution to software quality can be most directly made. This view of quality is aligned with the movement for Total Quality Management popular in manufacturing circles recently, in that it states that quality must be achieved through superior production processes rather than by later inspections or adjustments. Software specialists might have been exposed to a similar idea through the CMM (Capability Maturity Model) propounded by the Software Engineering Institute.

This research has shown that the Garvin framework has little effect on software quality priorities. Most stakeholders, regardless of their roles, believe that software quality can only be experienced through use, rather than through examination of program code or development methodologies.

REFERENCES

1. *Arthur L.J.* Measuring Programmer Productivity and Software Quality. — NY: John Wiley and Sons, 1985. — 292 p.
2. *Deutsch M.S., Willis R.R.* Software Quality Engineering: a Total Technical and Management Approach. — Englewood Cliffs. — NY.: Prentice Hall, 1988. — 317 p.
3. *Trienekens J.J.M.* et al. Entropy based software processes improvement // *Software Quality Journal*. — 2009. — **17**(3). — P. 231–243.
4. *Garvin D.A.* What Does «Product Quality» Really Mean? // *Sloan Management Review*. — 1984. — **26**(1). — 37 p.
5. *Kusters R.J., Trienekens J.J.M., Hassoldt W.* On the business impact of software process improvement // In *Proceedings 26-th Annual International Computer Software and Applications*. — USA: IEEE Computing Society. — P. 59–67.
6. *Nance K.L., Strohmaier M.* End user perception and software quality assessment // *Journal of International Information Management*. — 1997. — **6**(1). — P. 1–8.
7. *Wilson D.N., Hall T., Baddoo N.* A framework for evaluation and prediction of software process improvement success // *Journal of Systems & Software*. — 2001. — **59**(2). — P. 135–142.
8. *Kitchenham B., Pfleeger S.L.* Software Quality: The Elusive Target // *IEEE Software*. — 1996. — **13**(1). — P. 12–21.
9. *Haigh M.* Software quality revisited: Diverging priorities between stakeholder groups? // In *CIST.*, Drexel University: Philadelphia. — 2002. — P. 184.
10. *Haigh M., Verner J.* Examining Stakeholder Priorities for Software Quality Attribute Requirements // In *Proceedings of the International Workshop on Requirements Engineering for Business Need and IT Alignment, 29–30 August 2005*. — Paris: UNSW Press. — P. 85–92.
11. *Haigh M.* Research versus practice in software engineering: comparison of expert opinions to measured user priorities // *System Research and Information Technologies*. — 2009. — **8**(2). — P. 133–142.

Received 23.04.2010

From the Editorial Board: the article corresponds completely to submitted manuscript.