International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 03, March-2015

Information System Quality: State of the Art and New Model

Sarah Aouhassi, Mostafa Hanoun
Laboratory of Information Technology and modeling
Faculty of Sciences Ben M'Sik, University of Hassan II Casablanca
B.P. 7955 Sidi Othmane, Casablanca, Morocco

Abstract—**Information System (IS) becomes a big priority of organization for the majority of firms and government institutions. Among the main reasons of using information system, we can mention information availability or reliability, better data circulation or communication, and finally insure information visibility and ease of access. For the reasons above, all information system's components, namely human resources, hardware, software, procedures and data, must have a definite level of quality. The review of literature reveals that all existing models are limited to the software quality as a substitution of information system quality, in addition, almost all the surveys and studies done to measure the information system quality on different organizations are considering only the developers or the technical staff opinion and neglecting the managers, the users and the operating staff opinion. In this article, we will highlight the limits of existing models and propose a hybrid model integrating quality indicators measurements for all information system components; we will also give adapted surveys to each kind of information system player.**

Keywords— ***Information system, quality model, software quality, measurement indicators.***

I. Introduction

Quality is defined as a measure of excellence or a state of being free from defects, deficiencies and significant variations. It is brought about by strict and consistent commitment to certain standards that achieve uniformity of a product in order to satisfy specific customer or user requirements. ISO 8402-1986 standard defines quality as "the totality of features and characteristics of a product or service that bears its ability to satisfy stated or implied needs [4].

Information system is defined as an organized set of resources (human, software, hardware, procedures and data) which allow to collect, sort, classify, treat and transmit information on working environment.

The IS's missions is to improve communication, to structure information treatment, and to contribute to the storage and the management of data.

Information System Quality is obtained when all its components have a particular quality level, defined from a certain number of indicators from a relevant model.

This article's purpose is to define the five groups of indicators which give the quality measurement of human resource involved in IS, the software and application's quality, hardware's quality, procedure's quality and data's quality. The second section of this article is dedicated to the state of the art of IS quality model existing on the literature and focusing on their strengths and highlighting their weaknesses. The Third section is about defining indicators of each IS component. The fourth section provides the adopted model based on the five groups of indicators. The fifth section focus on the different types of surveys adapted to each kind of IS's actors followed by section 6 which contains conclusion and perspectives.

II. IS Quality model

A. *Selecting ISO 9126*

ISO 9126 is a Software Product Evaluation: Quality Characteristics and Guidelines for their Use-standard [3]. The standard is divided into four parts which address respectively the following subjects: Quality model, External metrics, internal metrics and quality in use metrics. ISO 9126 is an extension of previous work done by McCall (1977), Boehm (1978), FURPS etc. ISO 9126 specifies and evaluates the quality of a software product in terms of internal and external software qualities and their connection to attributes. It is composed of six general characteristics which define the global quality of an application: Functionality, Ability, Reliability, Efficiency, Maintainability, Usability, and Portability. Each one of these characteristics is decomposed on sub-characteristics.

B. *SquaRE*

Software Quality Requirements and Evaluation (SquaRE) [12], describes two distinguished models. The first is a quality model linked to the software's use and the second is a quality model specific to the software production. As the ISO 9126 standard, SquaRE follows the same principle with eight characteristics, decomposed on sub-characteristics:

Functional suitability (*Functional completeness, Functional correctness, Functional appropriateness*): degree to which a product or system provides the functions that meet stated and implied needs when used under specified conditions.

Performance efficiency (*Time behavior, Resource utilization, Capacity*): performance relative to the amount of resources used under stated conditions.

Compatibility (*Co-existence, Interoperability*): degree to which a product, system or component can exchange information with other products, systems or components and/or perform its required functions, while sharing the same hardware or software environment.

Usability (*Appropriateness Recognizability, Learnability, Operability*): degree to which a product or system can be used to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Reliability (*Maturity, Availability, Fault tolerance, Recoverability*): degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.

Security (*Confidentiality, Integrity, Non-repudiation, Accountability, Authenticity*): degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.

Maintainability (*Modularity, Reusability, Analyzability, Modifiability, Testability*): degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

Portability (Adaptability, Installability, Replaceability): degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

The SquaRE standard derive from ISO 9126 standard and redefine judiciously the software quality characteristics. The fact, for example, of having distinguished security as an integral characteristic or differentiate portability from compatibility make the model more relevant. However, it's difficult to apply because there is no link between characteristics and metrics. Beside, these standards define a general quality model which aims to qualify the software as a whole.

C. The McCall quality model

McCall software quality model, also known as Factors Criteria Metric (FCM) model [3], is made up of a selection of factors (the main eleven of fifty) representing an extern and a global view of quality. These factors are characterized by twenty three criteria which represents the intern view of quality: the programmer perspective. The last level of this hierarchical model is metrics (about 300). The quality factors describe different types of system behavioral characteristics, and the quality criterions are attributes to one or more of the quality factors. The quality metric, in turn, aims to capture some of the aspects of a quality criterion. These eleven quality factors have been distributed in three perspectives: product revision, product transition and product operations. Product revision perspective identifies the ability to change the software product. Product transition identifies the ability to adapt the software to new environments. Product operation perspective identifies the software fulfillment with its specification.

The model's purpose is to reduce the gap between users and developers by focusing on a number of software quality factors that reflect both the user's views and the developer's priorities.

The large number of indicators on this model leads to a substantial difficulty to apply on a real case.

D. *GQM (Goal-Questions-Metrics)*

GQM is an established and elaborated method for measurement in software engineering [2], it is an approach that defines a measurement model on three levels: Conceptual level (the Goal level), Operational level (the Question level) and Quantitative Level (the Metrics level). GQM have been defined as top-down model, so, measuring the quality can only start after the model has been completely specified, and the first results must wait until sufficient data has been collected. GQM includes a data structure, the so-called GQM tree, that helps identifying and interpreting metrics for a given measurement goal. Specific types of questions clarify certain aspects of the measurement goal.

E. QMOOD

The Quality Model for Object-Oriented Design (QMOOD) [16] is a hierarchical model derived from the ISO 9126 standard. There are four levels in QMOOD: Design Quality Attributes (functionality, effectiveness, understandability, extendibility, reusability and flexibility), Object oriented design Properties (inheritance, encapsulation, polymorphism, abstraction, coupling, cohesion, messaging, hierarchies, composition, design size, and complexity), Object oriented design Metrics (Design Size in classes, Number of Hierarchies, Average Number of Ancestors, Number of Polymorphic methods, Class interface Size, Number of methods, Direct Class Coupling, Cohesion Among Methods of Class, Measure of aggregation, Measure of Functional Abstraction, Data Access Metric) and Object oriented design Components (attributes, methods, objects (classes), relationships and class hierarchies).

These high level attributes are evaluated by using properties empirically identified and weighted. This model is conceived for object oriented applications and cannot be applied to other paradigm. Moreover, it only qualifies the program's conception: it doesn't consider the implementation quality or the respect of programming rules.

F. Factors Strategies

This model is based on detection strategy [15], which is defined as metrics based rules for detecting design flaws. The authors suppose that there is a current extensive use of metrics on quality software models, beside, metrics allow affirming that there is an anomaly in the code but they can't specify the cause of this anomaly. That's actually why they used a generic mechanism "Detection Strategy" for analyzing a source code model using metrics. The metrics, such as Weighted Method Count, Tight Class Cohesion or Access to Foreign Data, are submitted to the filtering and composition mechanism.

G. Source Inventory

This model is a continuous software quality supervision using a set of metrics in order to measure the source code quality while the developers are still writing the program [17]. Some of the metrics used in this model are total number of functions, total number of methods, bugs and dangerous constructs, memory handling problems, complexity problems, etc…

The Source Inventory model helps software developers, architects and managers to take control over their software's quality but it stays a low level model without an Information system quality overview.

H. SQALE

The SQALE method -Software Quality Assessment based on Lifecycle Expectations- is a model which translates the requirements applicable to the software and its source code over its lifecycle [6, 7].

The SQALE method is particularly devoted to the management of the Technical Debt of software developments. It allows to define clearly what creates the technical debt, to estimate correctly this debt, to analyze this debt upon technical and business perspective and to offer different prioritization strategies allowing establishing optimized payback plan.

The quality model SQALE is a hierarchical model based on the principles below: three levels from the most general (the characteristics) to the more detailed (requirements, control points). It contains rules that are used for normalizing the measures and the controls relating to the code, and other rules for aggregating the normalized values.

This model was set up to quantify the code's quality and evaluate the remediation effort. It's particularly adapted to developers for whom it was been conceived. However, it doesn't consider the functional requirements.

I. The Squale model

The squale model, as the FCM model (McCall model) and the ISO 9126 standard, is based on a hierarchical structure adopting a top-down approach [9]. It is based on available measurements on a specific moment to compute high level quality marks in its next level. The Squale model starts from raw data that it aggregates to give a quality measure at a more general level. This approach gives a sense to raw measures that are only comprehensive and readable by the technical staff. It permits as well to make sure that the quality computations are always based on concretes and identifiable measures. Even more, adopting such approach allows to obtain an image of quality earlier and to follow its evolution during its life cycle.

Thus, the Squale Model is composed of four levels divided into two groups.

High-level marks:

- A ***factor*** represents the highest quality assessment to provide an overview of project health. It is addressed to non-technical persons (users and customers) and based on factors of the ISO 9126 model.

- A ***criterion*** assesses one principle of software quality. It is addressed to managers as a detailed level to understand more finely project quality. The criteria used in the Squale model are tailored for the assessment of quality in information systems.

- A ***practice*** assesses the respect of a technical principle in the project. It is directly addressed to developers in terms of good or bad property with respect to the project quality. Good practices should be fulfilled while bad practices should be avoided. The overall set of practices expresses rules to achieve optimum software quality from adeveloper's point of view. Around 50 practices are already defined, based on Air France-KLM quality standards. However, the list of practices is open and such practices can be adjusted.

Low-level marks:

- A ***measure*** is a raw information extracted from the project data. Measures provide raw metrics which are used to compute high-level marks.

III. IS QUALITY INDICATORS MEASURMENTS

The previous models consider that the information system quality is equivalent to the software quality. Well, the software is just one of the five components of the whole information system [1]. Besides, the quality measurement is based only on the developers' opinion [14] without considering the other intervening opinions like managers, functional staff and users.

The adopted model is a hybrid one, constructed from the existing models. It will expand the study about the IS quality by giving a multitude of point of view.

In the next part of this paper, we present the definitions of IS quality indicators measurements and their calculating methods.

A. Human resources quality

We must distinguish four human resources categories involved in IS: managers, technical staff, functional staff and users.

1) Manager experience

The IS quality is directly affected by the IS manager experience [5]. Decisions and strategies adopted are determining for the whole IS intervening. This indicator will be measured in term of years of experience on the same job or on a similar one, and in term of competence [8] degree through diploma and certificates obtained in IS management.

2) Staff numbers involved in IS

This number is including every one that contributes directly or indirectly on IS development, such application developers, software administrators, procedures and maintenance responsible …etc.

3) IS staff experience

The experience accumulation of IS staff lead to a better quality of IS itself through avoiding frequent errors and reducing task's length. This indicator will be measured in term of years of experience and competence degree of each member of IS staff.

4) Users implication degree

We must distinguish intern user from extern user in their relation with IS. This indicator is measured by the number of interaction with available applications and software.

5) Resistance to change of users

The adherence degree of users facing the new practices related to IS.

6) User competence

The competence level of users affect the IS quality though their responsiveness to applications, software and hardware.

B. Hardware quality

1) Average duration of life

The difference between the acquisition date of hardware and when it becomes outdated (computer, printer, server…).

2) Rate of daily use

The number of hours past at using IT equipment divided by the number of daily work hours.

3) Budget allocated to hardware

The proportion of the annual budget spent on new hardware.

C. Software and application quality

1) Ease of use :

Software and applications have to be easy to use by the final users. This indicator is staggered from « 0 »: « difficult to use » to « 5 »: « too easy to use »

2) The code development maintainability

This indicator is measured by a yes/no question: Has the code been reused for other applications?

3) Flexibility or adaptability

The ability of software and applications to satisfy similar needs to requirements originally specified.

4) Response time

The duration in seconds between the time the request is executed and the response time, this indicator will be calculated in a qualitative way: fast / slow.

5) Complexity

The difficulty level while programming and handling software and applications.

6) The application/software size

This indicator can be measured in different ways: functions number, code line number, interfaces number, software cost, total time spent on programming…etc.

7) Friendly interfaces

The interfaces should be practical and intuitive according to user's opinion. This indicator takes the values: yes/no.

8) Users specifications conformity

Developed applications or software have to match with the requirements initially specified, this indicator takes the values: yes/ partly /no.

9) Utility

The gap between the situations with and without the software, in terms of efficiency and work duration. This indicator is staggered from 0 « no utility» to 5 « very useful», -1 if the situation with the software is worse than the situation before.

10) Budget allocated to software and application

The proportion of the annual budget spent on new software and/or on application development.

D. Procedures quality

A procedure is a sequence of operations whose implementation and logical concatenation are defined in order to achieve a particular sequence; it must contribute to the speed and the disappearance of errors in everyday tasks. We could measure the quality of procedures by the quality of the documents and by their applicability degree.

1) Documentation

The documentation quality depends on the production quality of this documentation (archiving methods, destruction of obsolete documents ...etc.).

2) Applicability

The quality of the procedures depends on their applicability by the IS intervening. This indicator is staggered from 0: «not applicable» to 5: « totally applicable».

E. Data quality

1) Structure

Are the information organized and arranged in databases. It's a yes/no question.

2) Updating and back up

The interval after which the data are updated and saved for each database.

3) Lack of redundancy

The number of duplicated data found on each database.

4) Relevance

The number of objectives and expected results from existing data.

IV. MODELING

As seen at section 3 of this article, the measuring indicators of IS quality are divided in five groups related to the five components of the IS (human resource involved in IS, software and application, hardware, procedure and data). The IS quality modeling will be done by two stages.

The first stage is to establish a quality model for each IS component separately, then we will aggregate the five models concerning the five IS components on one single model that allows to define precisely variables affecting significantly the IS quality [19].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 03, March-2015

Let's consider the following models:

$$HRQ_i=\alpha_0+\alpha_1 MEx_i+\alpha_2 SNI_i+\alpha_3 StEx_i+\alpha_4 UID_i+\alpha_5 RCU_i+\alpha_6 UC_i+\varepsilon_{Ri} \quad (1)$$

$\alpha_0 \dots \alpha_6$: model parameters & ε_{Ri} : error term

$$HQ_i=\beta_0+\beta_1 ADL_i+\beta_2 RDU_i+\beta_3 BAH_i+\varepsilon_{Hi} \quad (2)$$

$\beta_0 \dots \beta_3$: model parameters & ε_{Hi} : error term

$$SAQ_i=\gamma_0+\gamma_1 EU_i+\gamma_2 CdM_i+\gamma_3 FAd_i+\gamma_4 RT_i+\gamma_5 Cx_i+\gamma_6 ASz_i+\gamma_7 FIt_i+\gamma_8 USC_i+\gamma_9 Ut_i+\gamma_{10} BAS_i+\varepsilon_{Si} \quad (3)$$

$\gamma_0 \dots \gamma_{10}$: model parameters & ε_{Si} : error term

$$PrQ_i=\delta_0+\delta_1 Doc_i+\delta_2 Apl_i+\varepsilon_{Pi} \quad (4)$$

$\delta_0 \dots \delta_2$: model parameters & ε_{Pi} : error term

$$DQ_i=\eta_0+\eta_1 Str_i+\eta_2 UpBp_i+\eta_3 LR_i+\eta_4 Rl_i+\varepsilon_{Di} \quad (5)$$

$\eta_0 \dots \eta_4$: model parameters & ε_{Di} : error term

$$ISQ_i= \rho_0+\rho_1 RH_i+\rho_2 HQ_i+\rho_3 SQ_i+\rho_4 PQ_i+\rho_5 DQ_i+\varepsilon_{Ii} \quad (6)$$

$\rho_0 \dots \rho_5$: model parameters & ε_{Ii} : error term

Where:

ISQ	Information System quality
HRQ	Human resources quality
MEx	*Manager experience*
StNI	*Staff numbers involved in IS*
StEx	*IS staff experience*
UID	*Users implication degree*
RCU	*Resistance to change of users*
UC	*User competence*
HQ	Hardware quality
ADL	*Average duration of life*
RDU	*Rate of daily use*
BAH	*Budget allocated to hardware*
SAQ	Software and application quality
EoU	*Ease of use*
CDM	*The code development maintainability*
FAd	*Flexibility or adaptability*
RT	*Response time*
Cx	*Complexity*
ASz	*The application/software size*
FIt	*Friendly interfaces*
USC	*Users specifications conformity*
Ut	*Utility*
BAS	*Budget allocated to software and application*
PrQ	Procedures quality
Doc	*Documentation*
Apl	*Applicability*
DQ	Data quality
Str	*Structure*
UpBp	*Updating and back up*
LR	*Lack of redundancy*
Rl	*Relevance*

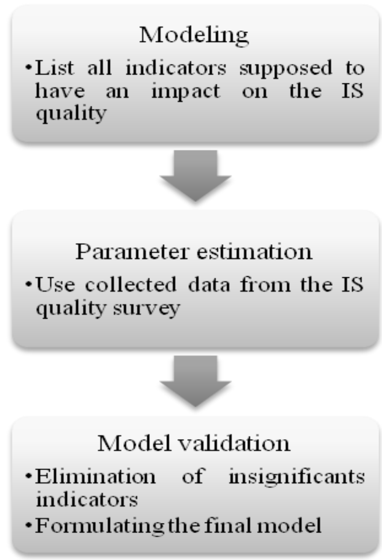

Figure 1 : Stages of modeling the IS quality

V. SURVEYS

The surveys are designed in order to be adapted for each type of the questioned: managers, technical staff, functional staff and users. The survey first part, regardless of type, helps to make a profile picture of the respondent, the second part deal with IS generalities, e.g. the IS department size, in numbers and staff skills or qualification. The third part emphasize the relationship between the respondent and other IS contributors, like the difficulty met when detailing technical requirements by managers for developers. The last part of the survey is about measuring indicators concerning software/application and hardware utilization in order to see if there is a way to optimize available resources, beside software and application impact on reduction time on performing a given task and on IS contributors' efficiency. The structure above is common [4] to the four types of survey; nevertheless every survey has its own distinctive feature specific to the different kind of staff, subject of the inquiry.

The survey addressed to manager focuses on the governance side of information system like the allocated budget for IS structure, the global strategies or orientations of the firm. The one addressed to developers concentrates on technical side of IS as the details of algorithms and functions composing the program or the application. The surveys addressed to functional team and users are more oriented at interfaces and the usability of available software and application.

VI. CONCLUSION AND PERSPECTIVES

The indicators measurement of the IS quality presented in this article have the particularity of being specific to each one of the five components of the IS (human resources, hardware, software, procedures and data), not just software, as described in almost all research already made. These indicators may be used later as standards for IS quality measures.

The survey adaptation and subdivision according to the four kinds of respondents will expand the IS quality perception by confronting the internal perception of IS contributors directly involved in the technical and organizational side, with the external perception of the different users.

The model adopted above will be tested with data from the different surveys applied to Moroccan universities. The collected data will allow defining all parameter and thereafter reducing the number of indicators in order to leave only significant ones on the final model. The IS quality model will help to diagnose the IS quality of an organization and provide a quality level with only limited number of indicators.

REFERENCES

[1] A. RIVET, "Normes de qualité et Système d'informations", CERMAV CNRS

[2] C. Gencel, K. Petersen, A. Ahmad Mughal, M. I. Iqbal "A decision support framework for metrics selection in goal-based measurement programs: GQM-DSFMS" The Journal of Systems and Software 86 (2013) pp. 3091– 3108

[3] C. Morley, "Management d'un projet système d'information : principes, techniques, mise en œuvre et outils" 5ème édition, novembre 2006.

[4] D. DURET, M. PILLET, " Qualité en production : de l'ISO 9000 à Six Sigma" Éditions d'Organisations, 3ème édition 2005.

[5] G. Gable, "Strategic information systems research: An archival analysis", Journal of Strategic Information Systems 19 (2010) pp. 3–16.

[6] J.L. Letouzey, M. Ilkiewicz, "Managing Technical Debt with the SQALE Method", special issue of IEEE Software inspearit, Arcueil, France, 2012.

[7] J.-L. Letouzey, Th. Coq, "The SQALE Analysis Model - An analysis model compliant with the representation condition for assessing the Quality of Software Source Code", VALID 2010,Nice, August 2010.

[8] J.Pepparda, J.Wardb, "Beyond strategic information systems: towards an IS capability", Journal of Strategic Information Systems 13 (2004) pp.167–194.

[9] K. Mordal, J. Laval, S. Ducasse,"An empirical model for continuous and weighted metric aggregation", Évolution et rénovation des systèmes logiciels Hermès (2011).

[10] [N. Gorla, S.Che Lin, " Detreminants of software quality: A survey of information systems project managers", Information and Software Technology 52 (2010), pp. 602 610.

[11] N. Gorla, T. M. Somers, B. Wongc, "Organizational impact of system quality, information quality, and service quality" Journal of Strategic Information Systems 19 (2010) pp. 207–228.

[12] N. Lai-Duc, "Software quality requirements and evaluation", LARION Computing, Dec 08, 2012.

[13] P. Besson, F. Rowe, "Strategizing information systems enabled organizational transformation: A transdisciplinary review and new directions", Journal of Strategic Information Systems 21 (2012) pp.103–124.

[14] P. DÉTRIE, « Conduire une démarche qualité, Éditions d'Organisations », 4ème édition 2003.

[15] R. Marinescu, "Detection strategies: Metrics-based rules for detecting design flaws". In 20th IEEE International Conference on Software Maintenance (ICSM'04), pages 350–359, Los Alamitos CA, 2004. IEEE Society Press.Computer.

[16] S. Chawla "Review of MOOD and QMOOD metric sets", International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 3, March 2013 pp. 448-451.

[17] T. Bakota,Á. Beszédes,R.Feren, and T. Gyimóthy. "Continuous software quality supervision using source inventory and columbus." In Companion of the 30th international conference on Software engineering, ICSE Companion '08, pages 931–932, New York, NY, USA, 2008. ACM.

[18] Y. Merali, T. Papadopoulos, T. Nadkarni, " Information systems strategy: Past, present, future" Journal of Strategic Information Systems 21 (2012) pp.125–153.

[19] R. Bourbonnais, "Econométrie", 5ème edition, DUNOD