19th International Conference — Science, Technology and Innovation
Booklets

RENIECYT - LATINDEX - Research Gate - DULCINEA - CLASE - Sudoc - HISPANA - SHERPA UNIVERSIA - Google Scholar DOI - REDIB - Mendeley - DIALNET - ROAD - ORCID

Title: Evaluating the software environment quality characteristic in an ubiquitous system

Authors: CUEVAS-RASGADO, Alma Delía, GUTIERREZ-TORNES, Agustín Francisco and NIÑO-MEMBRILLO, Yedid Erandini

Editorial label ECORFAN: 607-8695
BECORFAN Control Number: 2022-01
BECORFAN Classification (2022): 131222-0001

Pages: 17
RNA: 03-2010-032610115700-14

ECORFAN-México, S.C.
143 – 50 Itzopan Street
La Florida, Ecatepec Municipality
Mexico State, 55120 Zipcode
Phone: +52 1 55 6159 2296
Skype: ecorfan-mexico.s.c.
E-mail: contacto@ecorfan.org
Facebook: ECORFAN-México S. C.
Twitter: @EcorfanC

www.ecorfan.org

Holdings
Mexico Colombia Guatemala
Bolivia Cameroon Democratic
Spain El Salvador Republic
Ecuador Taiwan of Congo
Peru Paraguay Nicaragua

Introduction

Due to the large number of US that emerge, several studies have been presented over methods and methodologies. These studies present checklists of the characteristics of these US as in (Spinola, Massolar, & Travassos, 2007). However, in this paper a quality characteristics evaluation within software environment is presented. An arborescent structure of the characteristics involved, its sub-characteristics, attributes, metrics, methods, objectives, and formula is shown in this paper.

Agenda

Introduction

Methodology

Results

Annexes

Conclusions

References

Introduction

Years 1977 and 1978, were those of McCall (McCall, Richards, & Walters, 1977) and Boehm (Boehm, et al., 1978) respectively, and induced the identification of characteristics (through a list of software properties) that prevailed regardless of its type. Then came the ISO/IEC 9126 (ISO/IEC, 2001) in 1991 in which six main software features are proposed, as well as a set of formulas that, through identified metrics allow the quality measurement of the software product. The IEEE 1061 (IEEE, 2009) also presents its three levels model, Criteria, Factors and Metrics, using a tree structure. Other models such as Humphrey (Humphrey, 1989) have also prevailed over time. Now ISO/IEC 25000 (ISO/IEC, 2014) presents a suite based upon previous standards such as ISO/IEC 9126 (ISO/IEC, 2001), ISO/IEC 12207 (ISO/IEC, 2008), ISO/IEC 14598 (ISO/IEC, 2001) and ISO/IEC 15504 (ISO/IEC, 2013), to name a few.

Mark Weiser is often recognized as the 'father' of ubiquitous computing, who already in 1988 expressed the following: "*ubiquitous environments are environments where the US technological elements are inserted in the daily tasks making the user-system interaction natural and uninhibited, providing access to resources without limitation of time, means of access or place* " (Weiser, Some computer science issues in ubiquitous computing, 1993).

Related works

There are some works that show advances on the evaluation of software qualities, such as in (Hamsah et. al. 2021) where comparative analysis of quality model metrics via a meta-metrics approach is done, its results presents that most of the metrics are not of definitive derivation, proving opportunity to have a more structured and defined measurement function. For example, in AQUARIUM (Carvalho et. al. 2018) five product quality features were applied: Performance Efficiency, Compatibility, Usability, Reliability and Security. This model was developed for Mobile applications. In our model mobile applications are mainly evaluated through the Mobility Characteristic. Our model describes six characteristics: Software environment, Information access invisibility, Mobile support, Communication interoperability, Context sensitivity and Customization.

Methodology

Tools analysis

Design of HECA method for Ubiquitous Systems

Model application

Results analysis

Tools analysis

It really has four levels and a tree structure. The first level, which is the most general, lists the characteristics, which are the properties that must be present in any US. Then, the second level, enunciates the sub-characteristics. They are composed by those qualities that conform the characteristic with which they are associated. The attributes are found in the third level that details the properties and are the subjects of measurement. In the fourth and last level are the metrics, which define the way in which each attribute will be measured qualitative or quantitatively.

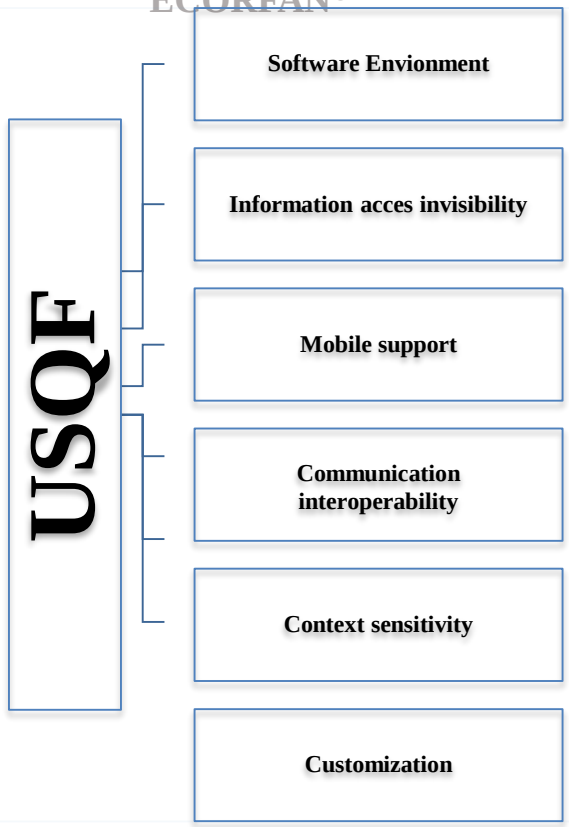


Figure 1 Ubiquitous Systems Quality Features (USQF) Source: Own.

Design of HECA method for Ubiquitou System

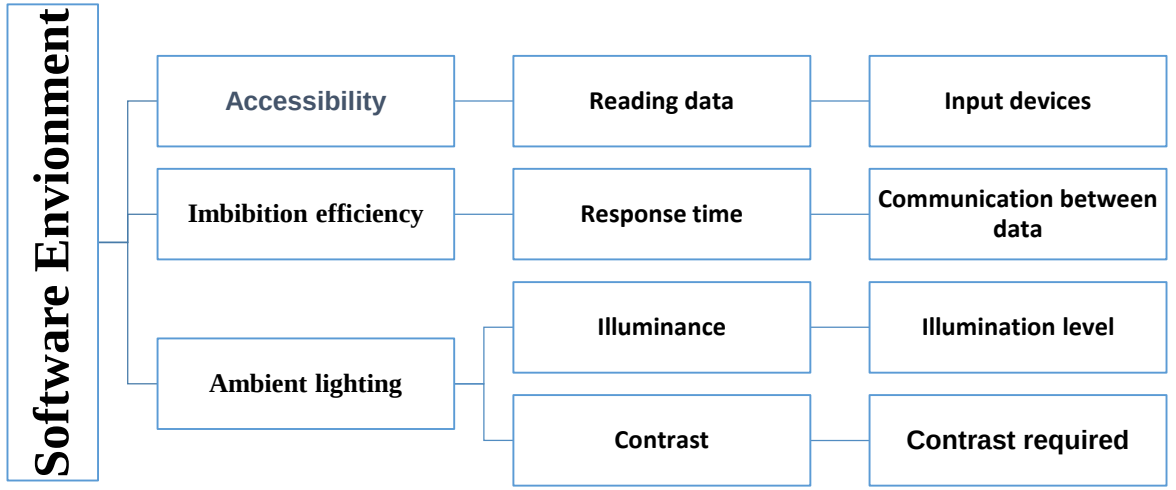


Figure 2. Software Environment quality characteristic Source: Own.

Design of HECA method for Ubiquitou System

ELEMENT TO EVALUATE	Sub-characteristic	Attribute	Metric
Accessibility	Reading data	Input devices	Number of everyday input devices used by the system:___ Total number of input devices used by the system:_____
Embedding efficiency	Response time	Communication between data	Number of times a request was made to the system within three seconds:_____ Number of total requests made (including those greater than three seconds):____
Ambient lighting	Illuminance	Illumination level	Maximum amount of lux measured during the use of a ubiquitous system:____ Allowed range of lux (500 <B<750):_____
	Contrast	Contrast required	Number of measurements where results are satisfactory:___ Total number of measurements performed:___

Model application

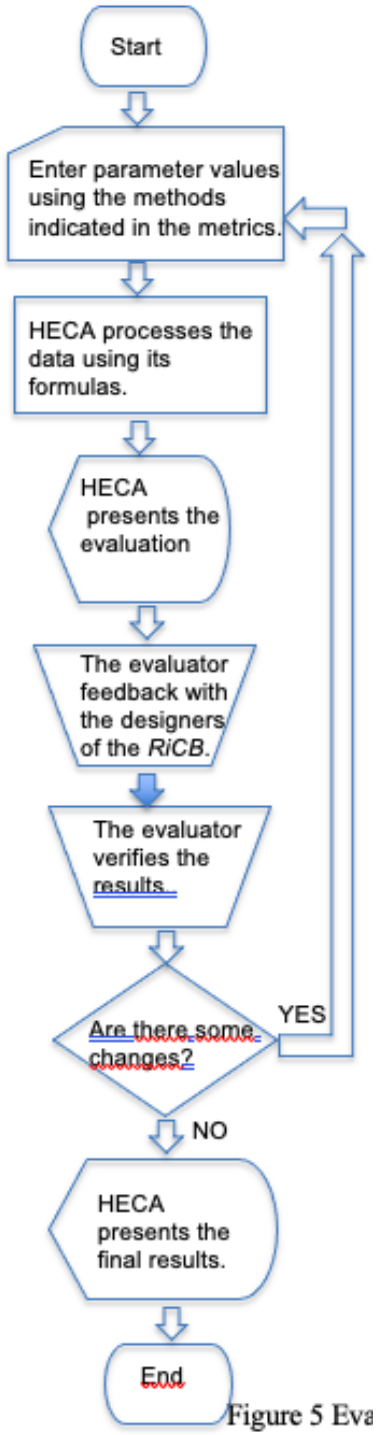


Figure 5 Eva

In evaluating the RiCB (Rescue in Collapsed Buildings) system quality the HECA model was used. Two evaluators and a Coordinator have made this assessment, in the Design process, Figure 5 shown this process.

Figure 5 Evaluation procedure using HECA (Source: Own).

Model application

Figure 6 presents the evaluation format used by HECA. Other numerical values of each metric are shown in the checklist. After using the formula, the system presents the value of the Software Environment characteristic, its sub-characteristics and its attributes.

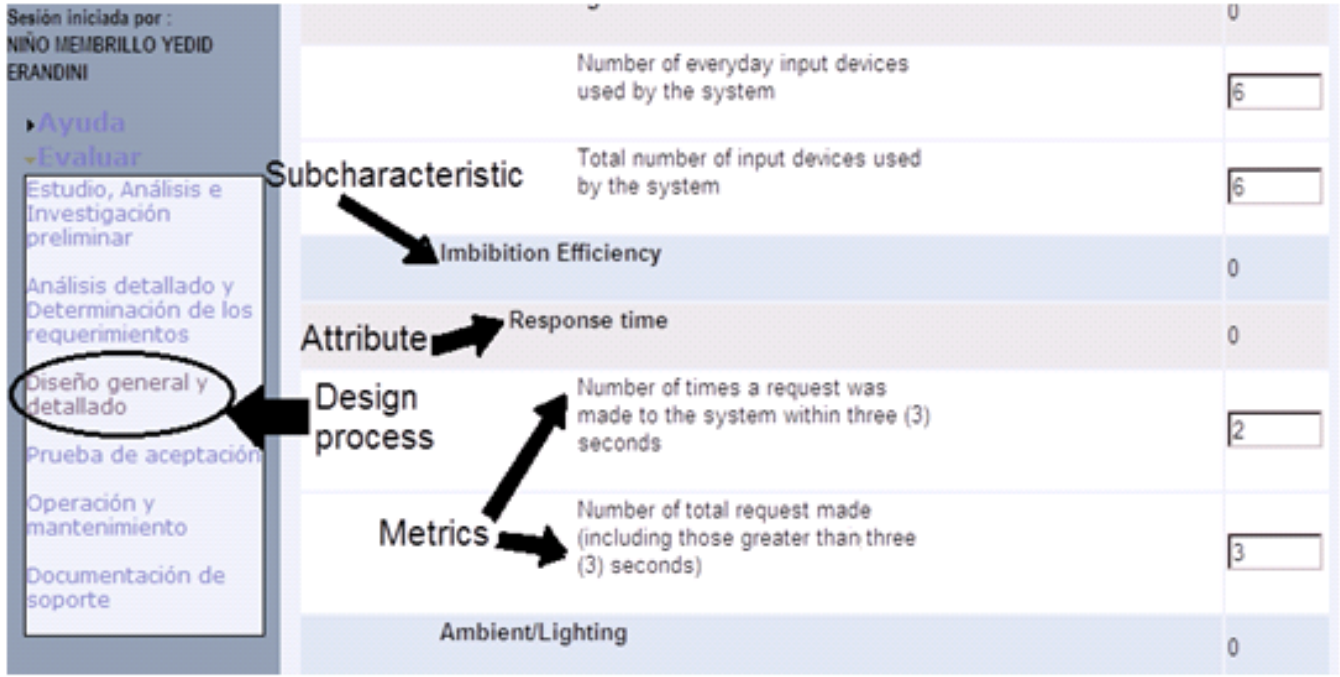


Figure 6. Evaluation of the Environment software in the Design process, before obtaining the result.

Result analysis

The scale conversion is then performed, from a zero to one or a zero to hundred in percentage to that from zero to three.

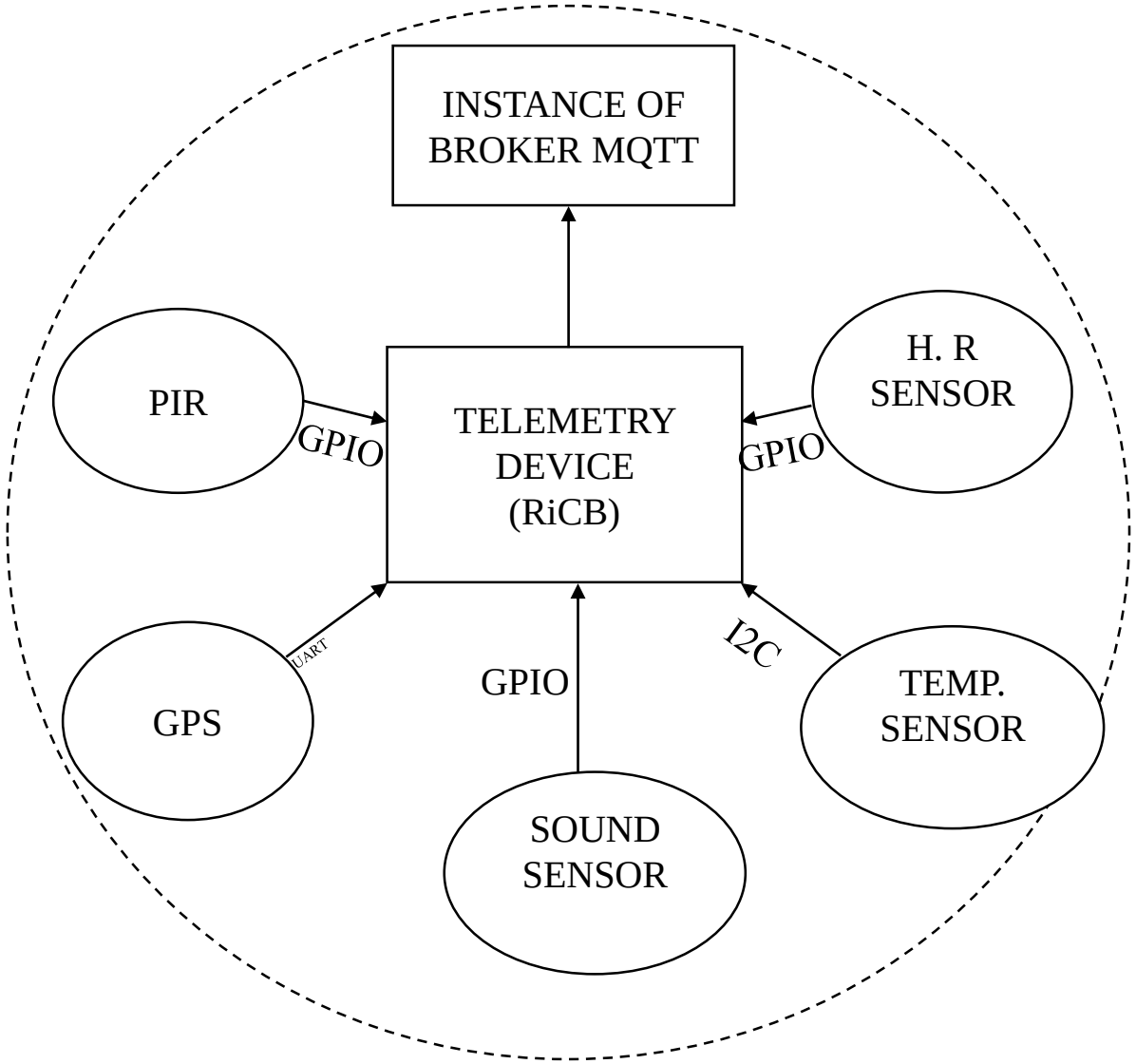
The result of the sub-characteristics and the Software Environment is presented in Table 2.

Accessibility	Imbibition Efficiency	Ambient/Lighting	Software Environment
2.3	2.5	2.2	2.3

Annexes

The RiCB (Cuevas-Rasgado et. al., 2022) Rescue in Collapsed Buildings is programed in a Raspberry pi zero, with movement sensor, temperature and GPS Neo-6m connected to it.

Figure 1 shows the general diagram of RiCB



Conclusions

The presented model was evaluated through the HECA software tool to which the Software Environment Feature values were added. With this tool it is possible to feed each metric with quantifiable values (numerical) according to the analysis and behavior of the projects to be evaluated. After the quantities are entered in Metrics, the model performs an average of the attributes to be assigned to the sub-characteristics and, in turn, an average of the characteristic to locate the software in a classification level of zero to three, where zero is the worst and three is the best.

Finally, the results of the evaluation of the Software Environment characteristic of an *RiCB* system is obtained. An RiCB systems objective is to search for victims in buildings collapsed by earthquakes. This system is considered ubiquitous because it uses several subsystems (user consultation system, Victim database administration system, tracking devices, public health institutions, shelters, interface system through mobile devices) and technologies (Raspberry Pi Zero card, thermal sensor, motion, location, and infrared), smartphone for consultation on tablets and smartphone all over the Internet.

Acknowledgements

This work has been funded by PRODEP-SEP grants number 511-6/2020-5931.

References

Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., & Merritt, M. (1978). *Characteristics of Software Quality.* Amsterdam, North Holland: TRW series of software technology.

Cortés, F. (2003). *Actualización y automatización de un modelo cualitativo para la evaluación de la calidad del software.* Tesis de maestría, Centro de Investigación en Computación del Instituto Politécnico Nacional, Mexico.

Cuevas, A. D. (2003). *La Evaluación de los Productos de Software.* Tesis de Maestría, Centro de Investigación en Computación del Instituto Politécnico Nacional, Mexico.

Cuevas, A. D., González, C. O., López, A., & Broeckl, U. (2021). Interoperability of Sensors in Buildings for Monitoring to Search odf Alive Victims after Earthquakes. *8th International Symposium on Language & Knowledge Engineering*. accepted to publish in special issues of Computing and System, CIC IPN, 2022.

Galico, D., Natanzon, K., Vega, C., Matalonga, S., & Solari, M. (2015). Context Sensitive Software: Definitions and Development of a Case Study in Gafas Google. *Research Paper. ISSN: 1688-6372*(13).

Humphrey, W. S. (1989). *Managing the Software Process.* Reading, MA: Addison-Wesley.

IEEE. (2009). *IEEE 1061-1998 (R2009) Standard for a Software Quality Metrics Methodology*. Retrieved from https://www.normadoc.com/german/ieee-1061-1998-r2009-2094004.html

References

SO/IEC. (2001). *ISO/IEC 14598-6: 2001, Software engineering - Product evaluation -Part 6: Documentation of evaluation modules*. Retrieved from https://www.iso.org/standard/24907.html

ISO/IEC. (2001). *ISO/IEC 9126-1,2001 Software engineering- Product quality- Part 1: Quality model*. Retrieved from https://www.iso.org/standard/22749.html

ISO/IEC. (2008). *ISO/IEC 12207: 2008, Systems and software engineering - Software life cycle processes*. Retrieved from https://www.iso.org/standard/43447.html

ISO/IEC. (2013). *ISO/IEC 15504-6:2013, Information technology - Process assessment - Part 6: An Exampler system life cycle process assessment model*. Retrieved from https://www.iso.org/standard/61492.html

ISO/IEC. (2014). *ISO/IEC 25000: 2014 Systems and Software Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE*. Retrieved from https://www.iso.org/standard/64764.html

Kray, C., Larsen, L. B., Olivier, P., Biemans, M., Bunningen, A. H., Fetter, M., . . . Vallejo, I. L. (2007). Evaluating Ubiquitous Systems with Users (Workshop Summary). In M. Mühlhäuser, A. Ferscha, & E. Aitenbichler (Eds.), *Constructing Ambient Intelligence AmI 2007 Workshops* (Vol. 11, pp. 63-74). Darmstadt, Germany: Springer.

References

McCall, J. A., Richards, P. K., & Walters, G. F. (1977). *Factors in Software Quality* (Vols. I-III). Rome, Italy: Rome Air Development Center.

Ruiz-López, T. (2014). *Un enfoque dirigido por modelos para el desarrollo de servicios para sistemas ubicuos basado en propiedades de calidad.* Tesis doctoral, Universidad de Granada.

Spinola, R., Massolar, J., & Travassos, G. H. (2007, 01 01). Checklist to Characterize Ubiquitous Software Projects. *XXI Simpósio Brasileiro de Engenharia de Software*. João Pessoa.

Tsang, T. M., Yeung, T. M., Chiu, D. K., Hu, H., Zhuang, Y., & Hu, H. (2010). Security Alert Management System for Internet Data Center Based on ISO/IEC 27001 Ontology. *2010 IEEE 7th International Conference on E-Business Engineering*, 178-183. Shanghai, China. doi:10.1109/ICEBE.2010.78

ECORFAN®