

La Sostenibilidad como Característica Transversal a la Calidad del Software

Leo Rosana*, Salgado Carlos⁺, Sanchez Alberto⁺, Peralta Mario⁺

* Departamento Académico de Ciencias Exactas, Físicas y Naturales. Universidad Nacional de la Rioja. La Rioja, Argentina
leorosana@gmail.com

⁺ Departamento de Informática Facultad de Ciencias Físico-Matemáticas y Naturales Universidad Nacional de San Luis
Ejército de los Andes 950 – C.P. 5700 – San Luis – Argentina
e-mail: {csalgado, mperalta, [asanchez](mailto:asanchez@unsl.edu.ar)}@unsl.edu.ar

RESUMEN

En la actualidad, es difícil pensar algún proceso de gestión que no aproveche las ventajas que brinda un software, en ese sentido la calidad del mismo se constituye en un punto importante a tener en cuenta en vista de la satisfacción de necesidades expresas o implícitas. Para tal fin es necesario utilizar modelos o estándares.

Por otro lado, se ha generalizado la idea de *sostenibilidad*, el color verde, la concientización por la preservación de los recursos del planeta, etc. incluso en el área de las Tecnologías de la Información y las Comunicaciones (TIC) y sobre todo en aspectos de infraestructura (centros de cómputos, servidores, dispositivos que consumen energía).

La sostenibilidad se analiza desde tres dimensiones: ambiental (para preservar la biodiversidad sin renunciar al progreso económico y social), económica (para lograr rentabilidad) y social (para lograr consenso social).

El presente trabajo pretende demostrar la importancia de incorporar como característica de calidad del software a la sostenibilidad, basado en estándares de calidad, como por ejemplo el modelo de calidad definido por la norma ISO/IEC 25010, pero teniendo en cuenta las guías y buenas prácticas que se correspondan con la sostenibilidad.

Palabras Clave: *Calidad de Software, Sostenibilidad, ISO 25010, Modelos de Calidad.*

CONTEXTO

El presente trabajo se enmarca en el Proyecto de Investigación: Ingeniería de Software: Estrategias de Desarrollo, Mantenimiento y Migración de Sistemas en la Nube – Facultad de Ciencias Físico-Matemáticas y Naturales, Universidad Nacional de San Luis. Proyecto N° P-03-2020. Dicho proyecto es la continuación de diferentes proyectos de investigación a través de los cuales se ha logrado un importante vínculo con distintas universidades a nivel nacional e internacional. Además, se encuentra reconocido por el programa de Incentivos.

1. INTRODUCCIÓN

La calidad de los sistemas informáticos se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones debido a que, cada vez más, sus procesos más importantes y, por lo tanto, la propia supervivencia de las organizaciones depende de los sistemas informáticos según se menciona en [1]. Cuando hablamos de sistemas informáticos, debemos mencionar que los

productos de software son un componente de importancia dentro del sistema que los contiene, y la calidad de éste afectará a la calidad del sistema en su conjunto. Aquí, entonces, sería importante dar alguna definición sobre calidad de software, si bien en la literatura hay varias definiciones, podemos mencionar la de [2] donde se define a la calidad de software como el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados, y de las características implícitas que se espera de todo software desarrollado profesionalmente. También en [3] se la define como: “Grado en que el producto software satisface las necesidades expresadas o implícitas, cuando es usado bajo condiciones determinadas”.

La especificación y evaluación de la calidad del software es factor clave para garantizar el valor a las partes interesadas. Esto se puede lograr mediante la definición de las características de calidad necesarias y deseadas asociadas con las metas y objetivos del sistema. Es importante que dichas características se especifiquen, midan y evalúen siempre que sea posible utilizando medidas y métodos de medición validados o ampliamente aceptados [4].

Para garantizar la calidad de software es importante implementar algún modelo o estándar que permita la gestión de atributos en el proceso de construcción de software, teniendo en cuenta que la concordancia de los requisitos y su construcción son la base de las medidas de calidad establecidas [5].

La ventaja de estos estándares es que la calidad se convierte en algo concreto, que se puede definir, medir y, sobre todo, planificar. Ayudan también a comprender las relaciones que existen entre las diferentes características de un producto

de software.

Si bien, la norma ISO/IEC 14001:2015 [6] proporciona a las organizaciones un marco con el que proteger el medio ambiente y responder a las condiciones ambientales cambiantes, manteniendo el equilibrio con las necesidades socioeconómicas, este estándar no incluye requisitos específicos para otros sistemas como lo son los de gestión de la calidad.

Por otro lado, si se habla de satisfacer necesidades, es importante tener en cuenta un término que se escucha cada vez con más frecuencia, como lo es *sostenibilidad*. Surge en el año 1987 en el Informe Brundtland [7], titulado “Nuestro Futuro Común”, y por la necesidad de estudiar y delimitar el impacto de las actividades humanas sobre el medio ambiente. Allí se define la sostenibilidad como “el desarrollo que satisface las necesidades del presente sin comprometer la habilidad de generaciones futuras de satisfacer sus propias necesidades”, todo ello garantizando el equilibrio entre el crecimiento económico, el medio ambiente y el bienestar social, pilares del término que se analiza.

La autora en [8] presenta revisiones sistemáticas a la literatura referida a la sostenibilidad en el ámbito del software. Realiza una cronología desde 1987 cuando la Comisión Mundial de Medio Ambiente y Desarrollo (CMMAD) expone [7]. Presenta los términos Green IT (Tecnologías de la información verdes) y Green IS (Sistemas de información verdes) con diversas definiciones y en el caso de Green IT, varias subdivisiones.

Afirma que “El proceso de desarrollo del software para ser sostenible tiene que hacer uso responsable de los recursos que consumen energía y usan materias primas no renovables. Y el software debe ser fácil de mantener”.

El involucrar a los equipos de desarrollo permitirá evaluar cuan sostenible es el proceso de construcción y el producto de software, como así también, tomar conciencia de las ventajas que implica dicho producto en la toma de decisiones todo vinculado a la reducción del impacto medio ambiental en relación al uso de las TIC.

En [9] se explica que la Sostenibilidad actualmente, es un aspecto fundamental en las TIC, apunta al cuidado de los recursos de nuestro planeta, aunque tiene más alcance que el factor ecológico si nos basamos en sus tres pilares fundamentales: Social, Económico y Ecológico (ambiental).

Al aplicar y evaluar criterios de sostenibilidad en cada elemento de un proyecto, se podrá establecer el grado o nivel de sostenibilidad (no concienciado, equitativo, viable, soportable, sostenible)

Valorar la sostenibilidad mediante una sencilla puntuación o medición en un rango acotado, permitirá tener una idea cuantitativa de la misma. Estableciendo un umbral se podrá determinar si el objeto de estudio es o no sostenible.

La Sostenibilidad será un criterio esencial en las organizaciones a fin de alinearse a los compromisos de preservación de los recursos naturales, contribuyendo a la toma de decisiones estratégicas, tácticas y operativas.

La Norma Internacional ISO/IEC 25010:2011 [4], una de las divisiones de la serie SQUARE, describe el modelo de calidad para el producto software, presentando características y subcaracterísticas de calidad, criterios a tener en cuenta al momento de la evaluación.

En el 2015 un manifiesto expone principios y compromisos vinculados con

el diseño sostenible [10]. Infiriendo que la sostenibilidad es un concepto sistémico, tiene múltiples dimensiones (social, medioambiental, económica, técnica y humana) y todas deben analizarse.

Al hablar de tecnologías sostenibles, se piensa en un menor consumo de energía, empleo de menor cantidad de recursos, la no contaminación, el reciclado o reutilización, siempre enfocados en satisfacer las necesidades de la sociedad. De allí se define un producto sostenible como aquel que aporta beneficios ambientales, sociales y económicos resguardando la salud pública, el bienestar y el medio ambiente en todo su ciclo de vida.

Bajo estos conceptos, se puede decir que el software también puede ser sostenible, cuando su desarrollo se basa en el uso adecuado de recursos y cuando su impacto negativo en la economía, la sociedad y el medio ambiente, es mínimo o, en el mejor de los casos resulta positivo respecto del desarrollo sostenible [11]. Pero, ¿cómo lo medimos?

Una medida es un valor asignado a algún atributo de dicho producto. Una métrica interpreta esos valores e incluso puede relacionarlos. El IEEE Standard Glossary of Software Engineering Terminology define métrica como “una medida cuantitativa del grado en el que un sistema, componente o proceso posee un atributo determinado”. Al evaluar las métricas, surgen los indicadores. Un indicador es una métrica o combinación de ellas que proporciona comprensión sobre el proceso, proyecto o producto software [2].

Los Modelos de Calidad del Software (MC), son artefactos específicamente diseñados y construidos para apoyar en estos procesos.

Según Calero et al. en [12], y el estándar ISO 8402 [13] un modelo de calidad puede definirse como el conjunto

de factores de calidad, y de relaciones entre ellos, que proporciona una base para la especificación de requisitos de calidad y para la evaluación de la calidad de los componentes software. Los modelos de calidad se estructuran generalmente como una jerarquía (ya sea un árbol, ya sea un grafo dirigido), donde factores de calidad más genéricos, como eficiencia o usabilidad, se descomponen en otros más particulares, como tiempo de respuesta o facilidad de aprendizaje, probablemente en diversos niveles de descomposición. También menciona que las propuestas existentes de modelos de calidad se pueden clasificar según si tienen un enfoque de modelos de calidad fijos, a medida o mixtos.

Entre los modelos de calidad fijos se pueden observar los de McCall et al. (1997) [14], Boehm et al. (1978) [15], Keller et al. (1990) [16] y FURPS Grady y Caswell (1987) [17]. Para los modelos de calidad a medida existen diversas propuestas de métodos para crearlos entre las que podemos destacar a GQM (Goal-Question-Metric) de Basili [18] y la del estándar IEEE 1061 [19].

Para el caso de los modelos de calidad mixtos se pueden destacar el ADEQUATE Horgan [20], el modelo de Gilb [21] y el modelo propuesto en el estándar ISO/IEC 9126-1 [22], este último es actualizado y reemplazado por el estándar ISO/IEC 25010 [4].

Coincidiendo con lo expresado en [10], la construcción de modelos de calidad viene dificultada por distintas circunstancias relacionadas con: (1) el equipo que realiza la construcción del modelo, en el caso de que este equipo no tenga experiencia en la construcción de modelos de calidad o bien en el contexto del dominio del componente objeto; (2) el dominio para el que se construye el modelo, para el que, en muchas ocasiones, no existe una terminología

común; (3) factores metodológicos, ya que es difícil conocer el nivel de profundidad hasta el que es necesario descomponer los modelos, y por tanto cuándo se puede decir que un modelo de calidad se ha finalizado.

Por lo expresado anteriormente, el objetivo de este trabajo es desarrollar un modelo que favorezca la evaluación de la calidad del software en una organización, con el fin de detectar los requerimientos de los usuarios y/o necesidades del dominio no cubiertas, en pos de mejorar la eficiencia organizacional a través de la mejora en la calidad del software que da soporte a los procesos de la organización teniendo en cuenta criterios de sostenibilidad.

El modelo que se propone se centrará en la calidad del producto de software, tomando como punto de partida el modelo conceptual de los procesos de la organización.

Es de gran importancia que las organizaciones puedan contar con modelos de calidad que partan del modelo conceptual de sus procesos y que les permita la evaluación objetiva y de manera sistémica de la calidad del software que da soporte a los procesos organizacionales, respetando y cumpliendo con los requisitos de sostenibilidad que las sociedades actuales exigen.

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Los principales ejes de trabajo de esta línea de investigación están asociados a:

- Evaluación de la calidad de productos de software.
- Estudio de modelos conceptuales con base en la sostenibilidad aplicados a la calidad de productos de software.
- Estudio de estándares y metodologías aplicadas a la construcción de MC.

3. RESULTADOS OBTENIDOS/ESPERADOS

La línea de I+D presentada en este trabajo se está desarrollando, tomando en cuenta los objetivos del Proyecto de I+D que la contiene.

De manera específica para esta línea se obtuvieron hasta la fecha los siguientes resultados:

- Lograr, a través de un MC, una evaluación objetiva de un determinado software para saber si este cubre las demandas organizacionales y las necesidades de los usuarios, y en función de ello proponer las mejoras necesarias para cubrir las necesidades no cubiertas por el mismo.
- Estudio de los modelos y normas de calidad aplicados a productos de software.
- Estudio de metodologías o métodos que guíen la construcción de modelos de calidad.

Los resultados esperados son:

- Concluir con la definición del modelo conceptual de calidad aplicado a productos software a través de la metodología y norma de calidad seleccionada, sin perder de vista los criterios de sostenibilidad.

4. FORMACIÓN DE RECURSOS HUMANOS

En esta línea de investigación se trabaja en lo referente a distintos modelos y métodos de evaluación de calidad. Se está trabajando en el Proyecto de Tesis de la Maestría en Calidad de Software (Plan Ord. 017/09-CD) de la Lic. Rosana Leo, de la Facultad de Ciencias Físico-Matemáticas y Naturales, Universidad Nacional de San Luis.

Como así también se están llevando a

cabo algunas tesinas de grado para la Licenciatura en Computación. Y trabajos finales de carrera de la Ingeniería en Informática e Ingeniería en Computación de la Facultad de Ciencias Físico-Matemáticas y Naturales, Universidad Nacional de San Luis.

5. BIBLIOGRAFÍA

- [1]. Piattini Velthuis et al., Calidad de sistemas de información. 2ª edición: RaMa, 2011.
- [2]. R. Pressman, Ingeniería del Software. 7^{ma} Ed: Mcgraw-Hill, 2010.
- [3]. ISO/IEC 25000 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-System and software quality models.
- [4]. ISO/IEC 25010:2011 Systems and software Engineering-Systems and Software Quality Requirements and Evaluation (SQUARE) System and software quality models
- [5]. Callejas-Cuervo, Mauro; Alarcon-Aldana, Andrea Catherine; Álvarez-Carreño, Ana María Modelos de calidad del software, un estado del arte. En: Entramado. Enero - Junio, 2017. vol. 13, no. 1, p. 236-250.
- [6]. ISO/IEC 14001:2015 Environmental Management Systems. Requirements with guidance for use.
- [7]. CMMAD. Comisión Mundial de Medio Ambiente y Desarrollo (1987). Informe Brundtland: "Nuestro futuro común".
- [8]. Dra. Anaisa Hernandez Gonzalez. "La sostenibilidad y el software"
- [9]. Montoya Guillermo - CEO Deiser S.L "Sostenibilidad del software: más allá de Green IT"
- [10]. Christoph Becker. Manifiesto Karlskrona. Sustainability design

- and software. 2015.
- [11]. Naumann - Dick - Kern – Johan. El modelo GREENSOFT: un modelo de referencia para el software verde y sostenible y su ingeniería
- [12]. C. Calero, M. Piattini, M. Moraga, Calidad del producto y proceso software: Ra-Ma, 2010.
- [13]. International Standards Organization (1986). ISO International Standard 8402: Quality Management and Quality Assurance-Vocabulary
- [14]. McCall, J.A., Richards, P.K. Y Walters, G.F. (1977) “Factors in Software Quality”. RADC TR-77-369, Vols I, II, III, US Rome Air Development Center Reports NTIS AD/A-049.
- [15]. Boehm.W., Brown, J.R., Kaspar, H., Lipow, M., Macleod, G.J. y Merritt, M.J. (1978). “Characteristics of Software Quality”. North Holland Publishing Company.
- [16]. Keller, S., Kahn, L. y Panara, R. (1990) “Specifying Software Quality Requirements with Metrics”. Systems and Software Requirements Engineering - IEEE Computer Society Press – Tutorial
- [17]. Grady R.B. y Caswell, D.L. (1987). “Software Metrics: Establishing a Company-Wide Program”. Prentice-Hall.
- [18]. Basili, V.R., Caldiera, G. y Rombach, H.D. (1994). “Goal Question Metric Paradigm”. En: Encyclopedia of Software Engineering 1, John Wiley & Sons.
- [19]. Institute Of Electrical Electronic Engineering (1998). IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology – Description.
- [20]. Horgan, G., Khaddaj, S. Y Forte, P. (1999) “Anessential Views Modelfor Software Quality Assurance”. En: Project Control for Software Quality, Shaker Publishing.
- [21]. Gilb, T. (1988). “Principles of Software Engineering Management”. Addison Wesley.
- [22]. ISO/IEC 9126-1. (2001). Software engineering — Product quality — Part 1: Quality model.