



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO DE SISTEMAS

CARRERA:

INGENIERÍA DE SISTEMAS

TEMA:

**“MÉTODOS MULTICRITERIO PARA LA PRIORIZACIÓN DE REQUISITOS
DE SOFTWARE”**

AUTOR:

ORNELLA ANDREA JURADO ANTÓN

TUTOR:

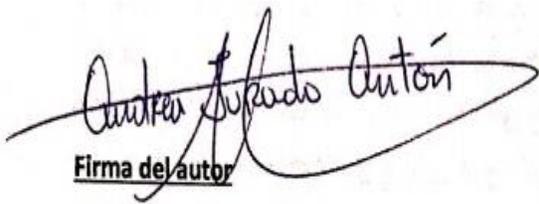
MSG. MIGUEL QUIROZ MARTÍNEZ

Abril 2021

GUAYAQUIL-ECUADOR

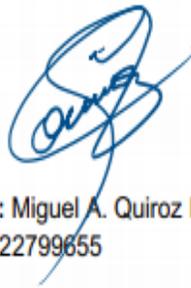
DECLARATORIA DE RESPONSABILIDAD

Yo, **Ornella Andrea Jurado Antón**, declaro que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad del/los autor/es.



Firma del autor

Nombre: Ornella Andrea Jurado Antón
CI. 0926585019



Firma:
(Tutor): Miguel A. Quiroz Martínez
C.I.: 0922799655

MÉTODOS MULTICRITERIO PARA LA PRIORIZACIÓN DE REQUISITOS DE SOFTWARE

Miguel Angel Quiroz Martinez^{1[0000-0002-8369-1913]} and Ornella Andrea Jurado Antón^{1[0000-0003-1941-9146]}

¹ Department of Computer Science, Universidad Politécnica Salesiana Sede Guayaquil, Ecuador, Chamber 227 y 5 de junio
mquiroz@ups.edu.ec, ojurado@est.ups.edu.ec

Resumen. El proceso heurístico de requisitos orientados a objetivos se utiliza para identificar los requisitos de software dependiendo de la necesidad de los stakeholders. Un sistema puede tener miles de requisitos, y es difícil priorizar una cantidad tan grande de requisitos cuando múltiples partes interesadas están involucradas en el proceso de inicio de requisitos. En la literatura, hemos determinado que cuando las preferencias de las partes interesadas se dan en forma de variables lingüísticas y se utilizan métodos de toma de decisiones de criterios múltiples en el proceso de adquisición, ciertos procesos existentes para adquirir requisitos orientados a objetivos apoyan la priorización de requisitos. Para mantener la calidad del sistema de software en el proceso de evolución y adaptarse a los atributos de calidad a medida que evolucionan los requisitos, debe existir una arquitectura de software clara que todos los desarrolladores deban comprender y se deben seguir todos los cambios en el sistema. La arquitectura del software se puede crear con anticipación, pero también debe actualizarse a medida que se actualiza el campo del software, por lo que los requisitos del sistema de software seguirán evolucionando. A menudo es difícil crear una arquitectura para un sistema o parte de un sistema de modo que la arquitectura cumpla con los requisitos de calidad requeridos. En el trabajo actual, se propone la práctica de un enfoque multicriterio para dar prioridad a los requerimientos de software, que emplean el uso de operadores de agregación jerárquica para obtener la calidad requerida del software. Los resultados obtenidos muestran que debido al mejoramiento para la toma de decisiones de ingeniería de software, la aplicabilidad del modelo de soporte de decisiones basado en la estandarización de información, la oportunidad y dar mayor importancia al tema. En términos de priorizar los requisitos, es fácil utilizar los métodos sugeridos y proporcionar un alto grado de flexibilidad para el éxito del proyecto de software.

Palabras Clave: métodos multicriterio, priorizar requerimientos; software de calidad; ingeniería de requerimientos; operadores agregación.

1. Introducción

En términos generales, especialmente en la ingeniería de software, el objetivo es lograr la calidad óptima del producto final. La calidad del software generalmente depende de la capacidad de cumplir con los atributos basados en información que satisfaga las necesidades de los usuarios que vayan a utilizar el software, y determinar estas capacidades como requisitos de software.

El éxito de un proyecto de software dependerá en gran medida de la correcta selección y estudio de requerimientos de software, resuelven la prioridad de los requisitos y los requerimientos de software y son una de las actividades más complejas. La base para la ingeniería de requisitos y el éxito del proyecto. Al respecto, los citados autores señalan que los proyectos suelen tener más requisitos y requerimientos que las limitaciones de tiempo y costo para ser implementados. En particular, el proceso de priorización incluye requisitos y actividades de requisitos para crear y mantener documentos del sistema relacionados. En esencia, el proceso de priorización es un proceso de aplicación de un método estructurado que estudia el sistema y realiza un grupo de gráficos idénticos para servir como especificaciones del sistema. Las actividades básicas de la ingeniería de requisitos se muestran en la Figura 1.



Figura 1. Actividades consideradas en ingeniería de requisitos.

Los requisitos que conocen la viabilidad deben ser verificados, porque los errores son muy costosos e implican cambios en todas las demás fases del proyecto de software, afectan los requisitos y conducen a una nueva priorización. Los usuarios deben visualizar el sistema en ejecución e imaginar cómo encaja con su trabajo para poder verificarlo. Las técnicas de verificación comúnmente utilizadas son:

- Comprobación de requisitos
- Realizar prototipos
- Crear casos de prueba
- Estudio de consistencia automático

La revisión involucra a todas las partes interesadas y debe verificar:

- Coherencia y correctitud
- Requisitos Verificables
- Requisitos Comprensibles
- Trazabilidad
- Requisitos Adaptables

Los errores encontrados deben ser registrados de manera formal para su posterior discusión y análisis. Para realizar el proceso de priorización de requisitos se deben seguir los siguientes pasos, que se repiten en el ciclo de vida del proyecto:

- Seleccione uno o más criterios de prioridad
- Una o más partes interesadas asignan valor a los criterios seleccionados
- Resumir estándares para lograr el orden final de requisitos

Dado que el proceso de priorización es iterativo, participan múltiples participantes, se analizará cada iteración de cada requisito y se designará el valor final como el requisito a cumplir. Cuando se utiliza la tecnología de distribución de valor, en cada demanda, todos tienen un valor digital más el correspondiente valor de conciencia de la demanda.

Para asignar un valor de demanda a cada participante, se deben considerar las siguientes tres variables.

- Conocimiento personal sobre los requisitos, para ello considera estándares que serían todo tipo de conocimiento desde nulo hasta experto. A cada uno de estos niveles se le asignará un peso.
- Clasificación individual: considere la jerarquía de individuos dentro y fuera de la organización. Dado que existen múltiples niveles de jerarquía en cualquier organización, la distribución de peso efectiva vendrá dada por los diferentes pesos de cada nivel. Para los participantes fuera de lo planeado estos pueden ser clientes o desarrolladores, etc., la asignación tendrá un valor exacto entre todos los considerados por la estructura.
- El valor otorgado por el individuo es un valor de carácter entero que compete al siguiente grupo $\{-9, -8, \dots, -1, 0, 1, \dots, 9\}$. Si el valor no es positivo, por ende, los requisitos para poder ejecutar el proyecto y el éxito del sistema son negativos, por lo que la implementación del sistema no es suficiente para realizar esta función.

Es decir, todas las actividades que tienen que ver con la fase de obtener y negociar los requisitos deben cambiar para poder estar adaptados a la nueva realidad: programadores y clientes están repartidos de manera geográfica. En este caso, el espacio entre todos los participantes está en la fase de definir los requisitos es un requisito a considerar.

El proceso descrito no requiere que los participantes estén en la misma ubicación geográfica ni al mismo tiempo. Sí, además de evitar que los participantes no puedan participar, necesita un facilitador para ayudar a cada experto a asignar valor, trabajar con cada participante y utilizar las herramientas y métodos futuros por sí mismos. Tiene determinadas características y asigna un valor importante a determinados requisitos.

Cabe recalcar que en los primeros pasos de un proyecto que tiene como fin el desarrollo de software, los requisitos suelen ser inexactos. A medida que avanzaba el proyecto y aumentaba el conocimiento del producto, se explicaron los requisitos con más detalle.

La priorización de requerimientos es un proceso que se puede completar en cualquier etapa del ciclo de vida, lleva a cabo de acuerdo con las diferentes variables que imponen las necesidades y el contexto de la organización.

Priorización de requisitos para una adecuada calidad de software, los autores presentaron que priorizar requisitos es un procedimiento para la toma de decisiones, y en ocasiones se vuelve muy complicado, pues en este proceso se deben determinar las funciones necesarias incluidas en el producto final y están relacionadas con la calidad del proyecto. [1]. Expansión cuantitativa del método MoSCoW, se fundamenta en que, si bien todos los requerimientos tienen el mismo nivel de importancia, es necesario resaltar los requisitos que pueden aportar mayor mérito al sistema para que el trabajo de desarrollo se pueda enfocar de método más efectivo. La diferencia de otras tecnologías en que la escala utilizada tiene un significado inherente, por lo que el usuario responsable de asignar la prioridad conoce los efectos reales de su decisión. [2]. En el tercer artículo que revisamos, los autores presentaron que los profesionales de software están involucrados en difíciles selecciones que requieren distintos puntos de vista. Un caso en particular es el desarrollo de priorización de requerimientos. Se utiliza para determinar

cuál requisito de software formar una reiteración específica desde de un conjunto de requerimientos candidatos. Todos los estándares involucrados en este proceso pueden tener diferentes propiedades, por lo que deben evaluarse en diferentes campos y los resultados deben mostrarse en el dominio lingüístico. [3]. En el cuarto artículo revisado, los autores presentaron que las diferentes tecnologías de razonamiento automático pueden respaldar la priorización de requisitos de estándares múltiples, con el objetivo de reducir el trabajo manual y mejorar la calidad de los resultados de clasificación de los requisitos de los candidatos. [4]. Multi-criteria decision making using ELECTRE, los autores proponen un modelo de soporte de decisiones basado en red, que utiliza ELECTRE como método de clasificación de prioridades. ELECTRE es un modelo de toma de decisiones multicriterio, que ha demostrado clasificar eficazmente varios problemas de toma de decisiones. El sistema propuesto utiliza un método de 100 puntos para obtener información de múltiples partes interesadas. [5]. Un breve resumen del quinto artículo revisado, los autores presentaron la comparación de dos métodos de toma de decisiones multicriterio (ADM) y AHP y ELECTRE I, tratando de probar el método más factible para priorizar estudios de casos reales. [6]. En este artículo, los autores proponen un método para priorizar los casos de prueba mediante el uso de la tecnología Analytic Hierarchy Process (AHP). Para mejorar la viabilidad de este método en escenarios del mundo real, aplicaron AHP en un entorno difuso para que las variables difusas que requieren una cuantificación precisa se puedan utilizar para especificar valores estándar. [7]. Software requirements prioritization, los autores presentaron que las tecnologías existentes están severamente limitadas en términos de escalabilidad, falta de cuantificación y priorización de los participantes, consumo de tiempo, interdependencia de necesidades y la necesidad de una intervención humana altamente especializada. Estos hallazgos son muy útiles para que los investigadores y profesionales mejoren el nivel actual de tecnología y práctica. [8]. En el noveno artículo investigado nos dimos cuenta de que los autores propusieron utilizar las técnicas incluidas en la llamada informática blanda, incluida la hipótesis de conjuntos difusos, la disposición neuronal y las maneras de búsqueda heurísticas o metaheurísticas. Computación blanda puede proporcionar la flexibilidad necesaria para crear métodos y modelos que toleren inexactitudes, información y aproximación, características específicas del entorno de toma de decisiones de SI. [9].

2. Materiales y métodos o Metodología computacional

Con el fin de priorizar los requisitos de manera jerárquica a través de operadores de agregación con el fin de identificar conjuntos que sean lo suficientemente importantes para asegurar la calidad del software, se propone el siguiente conjunto de actividades:

- Criterios y requisitos de selección
- Obtener información
- Estandarización de valores
- Determinar el vector de pesos y agregación

La descripción detallada de cada actividad propuesta es la siguiente:

- Se selecciona los criterios y requisitos en esta actividad y luego eligen los criterios y requisitos a evaluar. Siendo $C = \{C1, C2, \dots, Ck\}$ con $k \geq 2$ los criterios a evaluar, y $R = \{R1, R2, \dots, Rj\}$ con $j \geq 2$ los requisitos.
- Obtener información sobre las preferencias de los tomadores de decisiones. Esta información representa una evaluación de cada necesidad basada en el estándar. El vector de utilidad se expresa de la siguiente manera: $V_j = \{vj1, vj2, \dots, vjn\}$, donde vjk , es la preferencia con relación al criterio Ck del requisito Rj . La valoración se da en el intervalo $[0,1]$, siendo 0 el peor valor y 1 el mejor valor.

$$\tilde{v}_{jk} = \frac{u_{k \max} - u_{jk}}{u_{k \max} - u_{k \min}}$$

- Normalización de valor: Los valores de los ítems prioritarios se normalizan considerando si son de tipo beneficio o tipo de costo. Siendo \tilde{v}_{kj} , Valor normalizado, que se calcula mediante la fórmula 2 para el estándar de tipo de beneficio.

$$\tilde{v}_{jk} = \frac{u_{k \min} - u_{jk}}{u_{k \min} - u_{k \max(2)}}$$

- Agregación: La función de agregación OAG: $[0,1]^n \rightarrow [0,1]$ Se obtiene mediante un proceso de polimerización jerárquico. Se utilizo WPM, empleando el modelo de puntuación de preferencia lógica (LSP)
Con base en los criterios anteriores, se utiliza el método de agregación jerárquica para este trabajo. Porque se adapta de manera más realista al proceso de priorización de requisitos para identificar las colecciones que son importantes para un exitoso proyecto de software. El empleo de operadores de agregación de manera jerárquica flexibiliza el método, y la posibilidad de obtener directamente las preferencias del decisor y su expresión en el vector de ponderaciones es otra ventaja de este método.

3. Resultados

Con el fin de determinar la prioridad de los requisitos para identificar el conjunto de claves, con el objetivo de obtener la calidad suficiente del software, se seleccionaron estándares relacionados con la dificultad técnica, costo y valor del proyecto entre los cuatro requisitos de software, que pertenecen al sistema de información en salud. Posteriormente, se realiza una evaluación para cada requisito de la norma seleccionada, tal como se evidencia en la Tabla 1.

Tabla 1. Evaluación

Requisito	Dificultad	Costo	Valor
R1	0.2	0.4	0.5
R2	0.8	0.2	0.6
R3	0.4	0.7	0.3
R4	0.3	0.6	0.7

El estándar relacionado con la dificultad técnica y el costo es el estándar de tipo de costo, que se normaliza de acuerdo con (4). El estándar de valor de tipo de beneficio se normaliza de acuerdo con (3). Los resultados normalizados se muestran en la Tabla 2.

Tabla 2. Normalización

Requisito	Dificultad	Costo	Valor
R1	1.00	0.67	0.80
R2	0.00	1.00	0.80
R3	0.71	0.00	0.00
R4	0.86	0.33	1.00

La Tabla 3 muestra la estructura de agregación jerárquica obtenida. Uso de operadores de agregación que puedan reflejar la simultaneidad del establecimiento de LSP

Tabla 3. Estructura

Entradas	Operador					Id del Bloque
	Id del bloqueo		Operador			
Costo	0,4	C--	Balance Costo / Beneficio	0.7	C-	Prioridad global
Valor	0,6					
Dificultad Técnica				0,3		

4. Discusión

Los resultados del resumen estándar permiten ordenar los requisitos. Se da por determinado que el método de prioridad es: R1> R4> R2ΣR3.

Los resultados obtenidos muestran que debido al mejoramiento para la toma de decisiones de ingeniería de software, la aplicabilidad del modelo de soporte de decisiones basado en la estandarización de información, la oportunidad y dar mayor importancia al tema. En términos de priorizar los requisitos, es fácil utilizar los métodos sugeridos y proporcionar un alto grado de flexibilidad para el éxito del proyecto de software.

5. Conclusiones

En esta investigación se propuso un método para la priorizar los requisitos con enfoque a los métodos multicriterio. Las actividades utilizadas son los criterios de selección, la obtención de información sobre las preferencias del decisor, la normalización de valores y la normalización de las preferencias finales normalizadas. Entre las principales ventajas de este método, es posible modelar la importancia de los estándares y la compensación.

Referencias

1. Quiroz Martínez, M., Jurado Antón, O., Plua Moran, D., & Leyva Vazquez, M. (2020). Priorización de requisitos para una adecuada calidad de software. Serie Científica De La Universidad De Las Ciencias Informáticas, 13(6), 135-145.
2. Del Sagrado, J., Del Águila, I. M., & Bosch, A. (2018). Expansión cuantitativa del método MoSCoW para la priorización de requisitos.

3. Betancourt-Vázquez, A., Pérez-Teruel, K., Ramírez-Domínguez, J., & Leyva-Vázquez, M. PRIORIZACIÓN DE REQUISITOS DE SOFTWARE MEDIANTE UN MODELO MULTI-EXPERTO MULTI-CRITERIO CON INFORMACIÓN HETEROGÉNEA.
4. Morales-Ramirez, I., Munante, D., Kifetew, F., Perini, A., Susi, A., & Siena, A. (2017, September). Exploiting user feedback in tool-supported multi-criteria requirements prioritization. In 2017 IEEE 25th International Requirements Engineering Conference (RE) (pp. 424-429). IEEE.
5. Mary, S. S. A., & Suganya, G. (2016). Multi-criteria decision making using ELECTRE. *Circuits and Systems*, 7(6), 1008-1020.
6. Fernandes, J. M., Rodrigues, S. P., & Costa, L. A. (2015, June). Comparing AHP and ELECTRE i for prioritizing software requirements. In 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (pp. 1-8). IEEE.
7. Tahvili, S., Saadatmand, M., & Bohlin, M. (2015). Multi-criteria test case prioritization using fuzzy analytic hierarchy process. In Tenth International Conference on Software Engineering Advances (ICSEA 2015), November 15-20, 2015, Barcelona, Spain.
8. Hujainah, F., Bakar, R. B. A., Abdulgaber, M. A., & Zamli, K. Z. (2018). Software requirements prioritisation: a systematic literature review on significance, stakeholders, techniques and challenges. *IEEE Access*, 6, 71497-71523.
9. Casanova, C., Chichi, M., Gabioud, M. L., Pereyra Rausch, F., Prado, L., Rottoli, G. D., ... & De Battista, A. (2020). Toma de decisiones multicriterio en problemas de la ingeniería de software utilizando computación blanda. In XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020, El Calafate, Santa Cruz).
10. Barbosa, P. A. M., Pinheiro, P. R., de Vasconcelos Silveira, F. R., & Simão Filho, M. (2017, April). Applying verbal analysis of decision to prioritize software requirement considering the stability of the requirement. In *Computer Science On-line Conference* (pp. 416-426). Springer, Cham.
11. Achimugu, P., Selamat, A., & Ibrahim, R. (2016). ReproTizer: A fully implemented software requirements prioritization tool. In *Transactions on Computational Collective Intelligence XXII* (pp. 80-105). Springer, Berlin, Heidelberg.
12. Gerogiannis, V. C., & Tzikas, G. (2017, September). Using Fuzzy Linguistic 2-Tuples to Collectively Prioritize Software Requirements based on Stakeholders' Evaluations. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics* (pp. 1-6).
13. Hujainah, F., Bakar, R. B. A., Al-Haimi, B., & Abdulgaber, M. A. (2018). Investigation of stakeholder analysis in requirement prioritization techniques. *Advanced Science Letters*, 24(10), 7227-7231.
14. Garg, N., Sadiq, M., & Agarwal, P. (2017). GOASREP: Goal oriented approach for software requirements elicitation and prioritization using analytic hierarchy process. In *Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications* (pp. 281-287). Springer, Singapore.
15. Khan, J. A., Rehman, I. U., Khan, Y. H., Khan, I. J., & Rashid, S. (2015). Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique. *International Journal of Modern Education & Computer Science*, 7(11).
16. Maccari, E. A., Martins, S. B., & Martins, C. B. (2015). Multi-criteria project prioritization in a professional master's program. *JISTEM-Journal of Information Systems and Technology Management*, 12(2), 393-414.17.