

Modelo matemático y procedimiento para evaluación por complejidad de los requisitos software.

Msc. Karina Pérez Teruel^{1*}, Msc. Maikel Yelandi Leyva Vásquez², Ing. Karel Fernandez Cedeño³, Msc. Sasha Valdes Jimenez⁴, Ing. Daimara Mustelie Sanchidrian⁵

1 Universidad de las Ciencias Informáticas, Cuba, karinapt@uci.cu

2 Universidad de las Ciencias Informáticas, Cuba, mleyvaz@uci.cu

3 Universidad de las Ciencias Informáticas, Cuba, kfernandez@uci.cu

4 Universidad de las Ciencias Informáticas, Cuba, sasha@uci.cu

5 Universidad de las Ciencias Informáticas, Cuba, dmustelie@uci.cu

Resumen. La Ingeniería de Requisitos constituye una base importante para la estimación de esfuerzo y tiempo en proyectos de desarrollo de software. Los requisitos del software aportan una medida significativa del tamaño del producto a desarrollar. La Universidad de Ciencias Informáticas (UCI) cuenta con un método de estimación, el cual, en aras de ganar en precisión, incorpora una segunda estimación luego de la etapa de requisitos para contar con una entrada más detallada técnicamente. Para ello los requisitos deben ser clasificados según la complejidad que aportan a la implementación del software. En el presente artículo se detalla un modelo matemático para la clasificación por complejidad de los requisitos de software y una herramienta diseñada a fin de facilitar el proceso de evaluación de los requisitos en los proyectos de desarrollo de software de la UCI.

Palabras claves: requisitos de software, estimación, evaluación de complejidad.

Abstract. Requirements Engineering provides an important basis for estimating time and effort in software development projects. Software requirements give a significant measure of product size to be developed. The University of Informatics Sciences (UCI) has an estimation method, which, in order to gain accuracy, incorporates a second estimate after the requirements phase to have more detailed technical input. This requirement shall be classified according to the complexity they bring to the software implementation. This article details a mathematical model for the classification of complexity in software requirements and a tool designed to facilitate the process of assessing the requirements on software development projects at UCI.

Keywords: software requirements, estimation, complexity assessment.

1 Introducción

En la actualidad las nuevas Tecnologías de la Información y las Comunicaciones (TIC) provocan cambios sustanciales en los procesos de negocio, productos y servicios y las organizaciones que desarrollan Sistemas de Información (SI) no pueden ni han estado ajenas a estos retos estratégicos y oportunidades de negocio [1].

A pesar del protagonismo de los SI en muchos sectores de la actividad humana, los proyectos de desarrollo de software no siempre culminan exitosamente. El Standish Group da seguimiento a un estudio iniciado en el año 1994 en el que realiza un análisis del estado de la industria de software [2, 3]. Los últimos resultados han sido publicados como extensión de su Estudio del Caos en el año 2009 y describen la situación de la industria de software a nivel global [4]:

- El 32% de los proyectos fueron exitosos.
- El 44% excedieron el cronograma y presupuesto y/o no incluyeron todos los requisitos planificados.
- El 24% fueron cancelados antes de su terminación y sus productos nunca fueron usados.

Los errores introducidos en la etapa de requisitos son altamente perjudiciales fundamentalmente porque obligan al equipo de desarrollo a realizar un arduo y costoso trabajo de reprogramación para corregirlos. Resulta de 68 a 110 veces más costoso corregir un defecto de requisitos encontrado cuando el software ya está operando de lo que costaría si el mismo defecto fuera encontrado durante la etapa de desarrollo de requisitos [5]. Un error, omisión o mala interpretación en los requisitos obliga a los miembros del proyecto a rehacer todo el trabajo hecho sobre la base de requisitos incorrectos. Todo esfuerzo que permita reducir errores en los requisitos y prevenir este esfuerzo malgastado resulta una acertada decisión.

En el presente trabajo se propone un modelo para la evaluación por complejidad de los requisitos software. A continuación se presenta el marco teórico abordando las temáticas de la estimación y los métodos multicriterio. Posteriormente se introduce el modelo propuesto y se desarrolla un caso de estudio. El artículo finaliza con las conclusiones y trabajos futuros.

2 Estimaciones .

Por otra parte, todos los equipos de desarrollo deben estimar el esfuerzo, tiempo y costo requerido para ejecutar sus proyectos. Una estimación es una predicción de algún aspecto del futuro, basada en conocimiento y suposiciones en un tiempo específico. Las estimaciones están sujetas a incertidumbre debido a que esta información es incompleta e imperfecta. Los jefes de proyecto y otros involucrados frecuentemente esperan estimaciones con alto grado de precisión en etapas tempranas del proyecto, lo cual no es practicable por la inevitable incertidumbre. Mientras más temprano se estime en el proyecto, menos precisión tendrá. Las estimaciones generadas a partir de

los requisitos de alto nivel lidiarán con más incertidumbre que las producidas a partir de especificaciones de requisitos detalladas [5].

El primer paso para la estimación del proyecto es la evaluación del tamaño del producto software. La estimación del tamaño puede estar basada en requisitos textuales, en modelos de análisis, en prototipos o en diseños de interfaz de usuarios. [6]. Generalmente los requisitos del software son el producto de trabajo del ciclo de vida que la mayor parte de los proyectos obtienen invariablemente por lo que se tomó como base para estimar el tamaño del software en esta investigación.

Dentro de los métodos más usados para estimar el tamaño del proyecto se destaca el método de Puntos de Función [7], el cual se obtiene utilizando una relación empírica basada en medidas cuantitativas del dominio de información del software y valoraciones subjetivas de su complejidad. Sus objetivos son:

- Medir lo que el usuario pide y lo que el usuario recibe.
- Medir independientemente de la tecnología utilizada en la implantación del sistema.
- Proporcionar una métrica del tamaño.
- Proporcionar un medio para la estimación del software.
- Proporcionar un factor de normalización para la comparación de distintos software.

El análisis de puntos de función se desarrolla considerando cinco parámetros básicos del sistema, estos son:

- Entradas externas.
- Salidas externas.
- Consultas externas.
- Grupos de datos lógicos internos,
- Grupos de datos lógicos externos.

Con estos parámetros se calculan los puntos de función sin ajustar. A posteriori se aplica un factor de ajuste que resulta de valoraciones subjetivas efectuadas a la aplicación que está siendo medida y de su entorno. La suma de los puntos de función sin ajustar y el factor de ajuste representan los puntos de función ajustados.

Estas estimaciones están sujetas a imprecisiones. A medida que se avanza en la definición de los requisitos se va alcanzando una mayor claridad en el alcance y complejidad del producto. Es por esto que una estimación basada en requisitos aporta mayor precisión para la planificación de proyecto. Para usar los requisitos como entrada a un método de estimación es necesario evaluar la complejidad que aportan al desarrollo de software. Esta constituye la motivación del presente artículo.

3 Métodos Multicriterio.

La toma de decisiones bajo múltiples criterios es una tarea subjetiva que depende de las preferencias de un grupo de decisores o expertos. Los modelos de toma de decisión multicriterio se relacionan con la estructuración del problema, modelado de las

preferencias, y la construcción y caracterización de diferentes modelos de agregación de criterios, así como la construcción de herramientas que le brinden soporte a los modelos [8].

Uno de los paradigmas principales son los modelos funcionales[9]. La función de valor agrega todos los criterios para dar una medida del valor V definida de manera tal que:

$$V(x) > V(y) \Leftrightarrow X > Y \quad (1)$$

$$V(x) = V(y) \Leftrightarrow X \sim Y \quad (2)$$

Donde $>$ y \sim denotan las relaciones de preferencia e indiferencia entre las opciones x y y respectivamente. Una función de valor puede ser expresada en formas diferentes dependiendo de la independencia entre los criterios [9]. Debido a su simplicidad, la más usada es la forma aditiva[10]. Estos modelos permiten ordenar las opciones y clasificarlas en grupos predefinidos.

Existen varias propuestas de utilización de métodos multicriterio en el análisis de requisitos [11-13]. Sin embargo no se han encontrado reportes de su utilización en el análisis de la complejidad de los requisitos [14].

4 Modelo para la determinación de la complejidad de los requisitos de software.

Cada organización desarrolladora de software tiene peculiaridades que hacen necesario determinar que componentes incluir y cuáles serían sus pesos específicos, para, sobre esta base, evaluar sus requisitos en cuanto a complejidad para el desarrollo. A continuación se presenta una propuesta de proceso a seguir por la organización, que guía los esfuerzos para conseguir este objetivo (Figura 1).

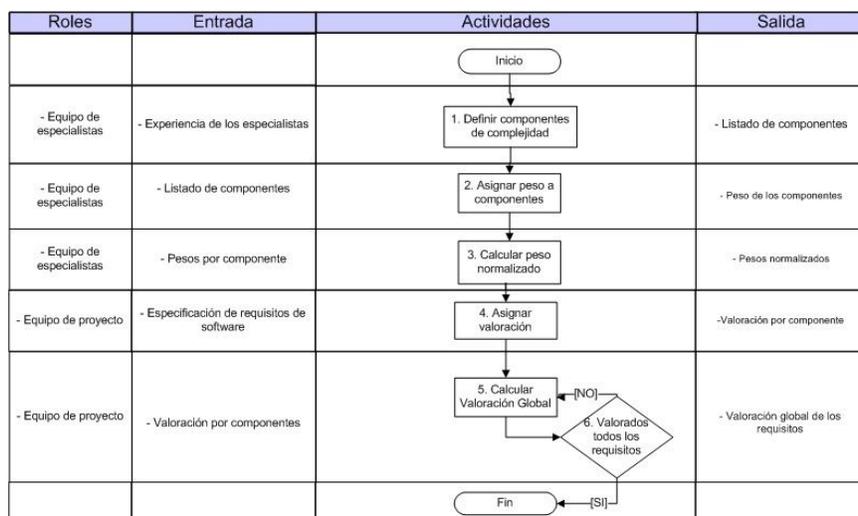


Figura. 1. Proceso de evaluación por complejidad de los requisitos en una organización desarrolladora de software.

Las actividades contenidas en el proceso se describen a continuación:

1. Se reúne un equipo de especialistas de la organización con experiencia en los desarrollos ejecutados en la misma, y determinan, sobre la base de las peculiaridades del desarrollo (software de gestión, software biomédico, software empotrado, etc.) de sus proyectos, cuáles serían los componentes de los requisitos que aportan complejidad para el desarrollo de los productos software.
2. El equipo de especialistas asigna pesos a los componentes de acuerdo a su incidencia en el esfuerzo necesario para el desarrollo.
3. El equipo de especialistas calcula mediante la fórmula (3) y usando como entrada los pesos determinados en la actividad 2, el peso normalizado de los componentes para facilitar los cálculos posteriores.
4. El equipo de proyecto, sobre la base de las definiciones logradas en las actividades 1, 2 y 3, asignan una valoración en cuanto a complejidad alta, media o baja a cada componente de los requisitos incluidos en su documento de Especificación de Requisitos de Software, de acuerdo a su proyecto específico. Estas valoraciones también tienen un valor numérico. Puede determinarse que uno o más componentes no aplican para el proyecto, en cuyo caso se asigna valor nulo.
5. El equipo de proyecto calcula, usando las valoraciones de los componentes, y mediante la fórmula (4), el Valor global de la complejidad de los requisitos incluidos en su Especificación de Requisitos de Software.
6. Si se no han calculado todas las valoraciones globales de los requisitos del proyecto volver a la actividad 5. Si se han terminado los cálculos ir a fin del proceso.

Las actividades 1, 2 y 3 se ejecutan a nivel organizacional y sus resultados son utilizados con posterioridad a nivel de proyectos como se describe en las actividades 4, 5 y 6.

Para la valoración se utilizó un modelo de scoring [15] utilizando una expresión algebraica que produce una puntuación para cada requisito, teniendo en cuenta los n componentes considerados más importantes por parte de los expertos. Para obtener esa valoración, cada uno de estos componentes es ponderado en relación a su importancia relativa con respecto al resto de los componentes. Se utiliza un modelo aditivo basado en la suma ponderada [10].

Se establece primeramente que cada requisito tiene un conjunto de componentes que aportan complejidad en mayor o menor grado a su implementación. Este aporte es representado por un peso que se le asigna a cada componente el cual se normaliza mediante la ecuación 3 con el objetivo de facilitar los cálculos posteriores.

$$w_n = \frac{C_n}{\sum_{i=1}^k C_i} \quad (3)$$

C_n Peso asignado al componente.

w_n Peso normalizado del criterio n

k Total de componentes

Por otra parte, como cada componente al tener incidencia, tiene una evaluación de complejidad Alta (A), Media (M) o Baja (B), se le asignó un valoración (v) de 1 a 3 a estas evaluaciones parciales. Este valor es multiplicado por el peso normalizado de cada componente y se obtiene una valoración global (V) del requisito n como se describe en ecuación 4.

$$V_n = \sum_{i=1}^k w_i v_i \quad (4)$$

V_n Valoración final obtenida por el requisito n .

v_i Valoración de cada componente del requisito n

A partir de la obtención de V_n se determina la evaluación en Alto, Medio o Bajo del requisito n :

- Baja si, $0 < V_n \leq 1.4$
- Media si, $1.4 < V_n \leq 2.4$
- Alta si, $2.4 < V_n \leq 3$

5 Caso de Estudio: Aplicación del modelo para determinar la complejidad de los requisitos en una organización desarrolladora de software.

En la Universidad de Ciencias Informáticas (UCI), se cuenta con un método de estimación dentro del programa de mejora. El mismo fue creado para ayudar a los proyectos a realizar las estimaciones de tiempo y costo, teniendo en cuenta un grupo de parámetros a partir de las actividades que suponga su realización, conociendo de

esta forma la implicación de las mismas en horas, meses y años y estimando finalmente según las tarifas horarias existentes el costo del proyecto.

El método en una primera versión, realizaba una estimación basada en paquetes funcionales, los cuales eran determinados a partir de los requisitos de alto nivel obtenidos en el proyecto técnico en la etapa de estudio preliminar. En una segunda versión del método, presentada en este artículo, se planteó la necesidad de realizar una segunda estimación más precisa luego de la etapa de requisitos para contar con una entrada al método con menos nivel de incertidumbre debido al grado de elaboración técnica de los requisitos lograda en esta etapa.

5.1 Actividad 1: Definir componentes de complejidad.

Para la determinación de estos componentes se escogieron un total de 5 especialistas vinculados a la producción, con participación en el Programa de Mejora (PM) desarrollado en la UCI y con experiencia en el desarrollo de requisitos de software, planificación de proyectos e implementación. Estos especialistas representaban las características y necesidades específicas de los tres centros vinculados al PM.

Previo a las primeras sesiones de trabajo los especialistas realizaron un estudio del estado del arte relativo a la clasificación por complejidad de los requisitos como entrada a métodos de estimación. Este estudio aportó ideas interesantes acerca de cuáles serían los componentes a incorporar en la solución teniendo en cuenta las necesidades específicas de la organización. Como base genérica para estimar el tamaño del software se tomaron los parámetros aportados por el método Puntos de Función.

Sobre esta base el grupo de especialistas en su primera sesión de trabajo en conjunto, determinó los componentes que interesaba incluir según las características y necesidades específicas de sus centros. Se tomaron a un conjunto de componentes comunes que aplicaban a todos los proyectos y fueron adicionados otros necesarios solo en algunos.

Los componentes identificados son los siguientes:

- Complejidad por Interfaces: El componente interfaz se aplica a requisitos que presenten algún tipo de complejidad en su interacción con los siguientes elementos, considerados subcomponentes y a los que se le aplica la misma forma de evaluación que a los componentes:
 - Humanas (formularios, informes)
 - Equipo (Tomógrafos, Rayos X); Ej API, drivers
 - Programación (Programas externos necesarios para apoyar el producto); Ej. Adicionar un nuevo usuario, usa el LDAP)
 - Comunicación (Protocolos de comunicación que serán utilizados); Ej. HL7
- Diferentes comportamientos: Un mismo requisito se comporta de manera diferente ante determinadas situaciones; Ej. Admisión de un paciente, puede ser normal, o por emergencia, en el primer caso recoge más información que en el segundo.
- Formas de inicialización: Un mismo requisito puede ser inicializado de diferentes formas, Ej. Ver detalles de una historia clínica, se inicializa cuando

creas la historia clínica en una vista previa, y cuando se selecciona mostrar historia clínica.

- Consultas a fuentes de almacenamientos: Los requisitos pueden presentar diversidad en la cantidad y complejidad de la interacción con la fuente de datos. Los posibles fuentes de datos identificadas son las siguientes, consideradas subcomponentes e igualmente evaluadas individualmente como componentes:
 - Base de Datos
 - Ficheros
 - Otros
- Restricciones de validación: Complejidad de todas las validaciones que lleve un requisito, tanto las validaciones de cara al cliente como en el servidor.
- Grado de reutilización: Complejidad de un requisito, para poder ser reutilizado por otros proyectos.
- Lógica de negocio: Los requisitos pueden presentar diferentes niveles de complejidad para la implementación de la lógica de negocio que contienen; Ej. operaciones, métodos matemáticos, CRUD, etc.

5.2 Actividad 2: Asignar peso a componentes.

En una segunda sesión de trabajo del grupo de especialistas procedió a asignar un peso de 1 a 5 a cada componente teniendo en cuenta el impacto que su incidencia en un requisito tenga para el esfuerzo de desarrollo de dicho requisito (Tabla 1).

5.3 Actividad 3: Calcular peso normalizado.

El peso total que puede tener un requisito si todos los componentes tienen incidencia es de 26. Tomando esta cifra total se llegó al peso normalizado (w) de cada componente usando la fórmula (3) (Tabla 1).

Tabla 1. Pesos asignados y normalizados de cada componente.

Interfaces	5	0.192307692
Diferentes comportamientos	2	0.076923077
Formas de inicialización	2	0.076923077
Consultas a fuentes de datos	5	0.192307692
Restricciones de validación	4	0.153846154
Grado de reutilización	3	0.115384615
Lógica de negocio	5	0.192307692
	26	1

5.4 Actividad 4: Asignar valoración a los componentes por requisito.

En aras de facilitar la determinación de la complejidad total de los requisitos se desarrolló una herramienta usando las aplicaciones para cálculo del paquete de Microsoft Office y de Open Office.

La herramienta cuenta con una pestaña que contiene una guía para los usuarios en la que se expone una explicación de cada componente identificado con ejemplos prácticos para su mejor entendimiento como se muestra en la Figura 2.

Componentes de Complejidad	
Para la determinación de la complejidad de los requisitos se analizan individualmente los componentes siguientes, llegando al resultado de si el requisito es de complejidad Alta, Media o Baja. La clasificación de la complejidad permite estimar el esfuerzo de implementación del requisito y contribuye a la decisión sobre la inclusión en las etapas de desarrollo del software.	
<ul style="list-style-type: none"> • Complejidad por Interfaces: El componente interfaz se aplica a requisitos que presenten algún tipo de complejidad en su interacción con los siguientes elementos: <ul style="list-style-type: none"> • Humanas (formularios, informes) • Equipo (Tomógrafos, Rayos X); Ej API, drivers • Programación (Programas externos necesarios para apoyar el producto); Ej. Adicionar un nuevo usuario, usa el LDAP) • Comunicación (Protocolos de comunicación que serán utilizados); Ej. HL7 	
<ul style="list-style-type: none"> • Diferentes comportamientos: Un mismo requisito se comporta de manera diferente ante determinadas situaciones; Ej. Admisión de un paciente, puede ser normal, o por emergencia, en el primer caso recoge más información que en el segundo. 	
<ul style="list-style-type: none"> • Formas de inicialización: Un mismo requisito puede ser inicializado de diferentes formas, Ej. Ver detalles de una historia clínica, se inicializa cuando creas la historia clínica en una vista previa, y cuando se selecciona mostrar historia clínica. 	
<ul style="list-style-type: none"> • Consultas a fuentes de almacenamientos: Los requisitos pueden presentar diversidad en la cantidad y complejidad de la interacción con la fuente de datos. <ul style="list-style-type: none"> • Base de Datos • Ficheros • Otros 	
<ul style="list-style-type: none"> • Restricciones de validación: Complejidad de todas las validaciones que lleve un requisito, tanto las validaciones en el lado del cliente, como en el servidor. 	
<ul style="list-style-type: none"> • Grado de reutilización: Complejidad de un requisito, para poder ser reutilizado por otros. 	
<ul style="list-style-type: none"> • Lógica de negocio: Los requisitos pueden presentar diferentes niveles de complejidad para la implementación de la lógica de negocio que contienen; Ej. operaciones, métodos matemáticos, CRUD, etc. 	

Figura. 2. Pestaña para la guía de evaluación de complejidad.

En la pestaña destinada al cálculo de la complejidad total de los requisitos (Figura 3) se diseñó una tabla que contiene primeramente una columna con el identificador y nombre del requisito, doce columnas para la evaluación parcial de los componentes que interfieren en esfuerzo de desarrollo del requisito con dos de ellos divididos en subcomponentes (Interfaz y Consulta a Bases de Datos) como se explicó con anterioridad, y una última columna para mostrar los resultados del cálculo de la complejidad resultante para cada requisito.

Requisitor	Componentar												Complejidad
	Interfazar				Diferente comportamiento	Formas de inicialización	Consultar fuentes de datos			Restricciones de validación	Grado de Reutilización	Lógica de Negocio	
	Humana	Equipar	Programación	Comunicación			Base de Datos	Ficheros	Otros				
RF1_Nombre del Requisito	A	A	A	A	A	A	A	A	A	A	A	A	Alta
RF2_Nombre del Requisito	B	B	B	B	B	B	B	B	B	B	B	B	Baja
RF3_Nombre del Requisito	M	M	M	M	M	M	M	M	M	M	M	M	Media
RF4_Nombre del Requisito	No	No	No	B	B	No	No	No	A	No	No	A	Media

Figura. 3. Pestaña para el cálculo de la complejidad total de los requisitos.

En el caso de los componentes Interfaz y Consulta a Bases de Datos que contienen subcomponentes, fue necesario realizar un cálculo adicional para promediar el resultado de las evaluaciones parciales de cada subcomponente respectivamente. En ellas se usaron los operadores proporcionados por las hojas de cálculo y la función COUNTIF como contador para promediar solo entre las celdas que contengan valores de A, M o B. Todas las celdas tienen en su único rango de valores posibles los caracteres A, M, B o No.

5.5 Actividad 5: Calcular valor global de la complejidad por requisito.

Por otra parte, se calcula según la fórmula (4) la sumatoria de la multiplicación del peso dependiendo del valor asignado de complejidad por el peso normalizado del componente lo que da el valor global de la complejidad del requisito.

Finalmente se clasifica el requisito en Alto, Medio o Bajo, según los rangos definidos en el modelo matemático, utilizando el valor global de la complejidad calculado anteriormente.

Para resumir los resultados obtenidos en el cálculo de la complejidad de los requisitos analizados se incorporó una tabla con contadores para determinar la cantidad de requisitos clasificados en cada grupo según muestra la Tabla 2.

Resumen	
Cant. de requisitos con complejidad Alta	1
Cant. de requisitos con complejidad Media	2
Cant. de requisitos con complejidad Baja	1

Tabla 2. Resumen de la cantidad de requisitos clasificados por complejidad.

Desde este momento se está usando los resultados obtenidos en la herramienta como entrada para el documento de Especificación de Requisitos de Software, que incluye en uno de sus campos la complejidad de los requisitos. Pero su uso fundamental, y la razón por la que fue creada, lo constituye la entrada a la segunda iteración del método de estimación, realizada luego de la fase de requisitos.

6 Conclusiones

El desarrollo de la investigación nos permitió arribar a las siguientes conclusiones:

- Es deseable para un proyecto realizar una reestimación luego de la fase de requisitos, que le permita, al ser hecha con datos más exactos, planificar con un mayor grado de confiabilidad.
- Luego de la etapa de desarrollo de requisitos se considera oportuno realizar esta reestimación contando como entrada con los requisitos obtenidos. Estos requisitos deben ser clasificados según su complejidad de implementación para refinar los datos a entrar en el método de estimación.
- Para clasificar los requisitos según su complejidad se estableció el modelo matemático que permitió calcular el Valor global de la complejidad de los requisitos partiendo de las valoraciones parciales asignadas a los componentes identificados y

un proceso guía para lograr este objetivo con actividades a nivel organizacional y de proyecto.

- Es oportuno el desarrollo de una herramienta para facilitar la aplicación del modelo en los proyectos productivos de la UCI. Para ellos se explotaron las facilidades que brindan las hojas de cálculo de Microsoft Excel y Open Office.
- La herramienta está siendo usada en los proyectos productivos de los tres centros de la UCI en el alcance del Programa de Mejora como entrada para la reestimación realizada luego de la etapa de requisitos y para el documento de Especificación de Requisitos de Software.

Como trabajos futuros se propone la utilización de la computación con palabras para la valoración de la complejidad y la utilización de operadores de agregación más complejos como los operadores OWA. Se propone la integración del procedimiento propuesto con un modelo para la priorización de requisitos. Se pretende elaborar posteriormente un Sistema de Soporte a la Toma de Decisiones en la ingeniería de requisitos basado en el modelo.

7 Bibliografía

1. Aurum, A. and C. Wohlin, *Engineering and Managing Software Requirements*. 2005, New York: Springer.
2. Chaos'01. *Extreme chaos. The Standish group report*. 2002 septiembre 2010]; Available from: http://www.standishgroup.com/sample_research.
3. Persse, J.R., *Process Improvement Essentials*. 2006: O'Reilly. 350.
4. Dominguez, J., *The Curious Case of the CHAOS Report 2009*. 2009.
5. Wiegers, K.E., *More About Software Requirements: Thorny Issues and Practical Advice*. 2006, Redmond: Microsoft Press
6. Wiegers, K.E., *Software Requirements, Second Edition*. 2003, Redmond: Microsoft Press
7. Albrecht, A.J. and J.E. Gaffney Jr, *Software function, source lines of code, and development effort prediction: a software science validation*. *Software Engineering, IEEE Transactions on*, 1983(6): p. 639-648 %@ 0098-5589.
8. Doumpos, M. and C. Zopounidis, *Preference disaggregation and statistical learning for multicriteria decision support: A review*. *European Journal of Operational Research*, 2010. **209**(3): p. 203-214.
9. Keeney, R. and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. 1993, Cambridge: Cambridge University Press.

10. Aznar, J. and F. Guijarro, *Nuevos métodos de Valoración. Valoración multicriterio*. 1ra Edición ed. 2005, Valencia, España.
11. In, H., D. Olson, and T. Rodgers. *A requirements negotiation model based on multi-criteria analysis*: IEEE.
12. Karlsson, J., C. Wohlin, and B. Regnell, *An evaluation of methods for prioritizing software requirements*. Information and Software Technology, 1998. **39**(14-15): p. 939-947.
13. Yamamoto, K. and M. Saeki, *Attributed goal-oriented analysis method for selecting alternatives of software requirements*. IEICE transactions on information and systems, 2008. **91**(4): p. 921-932.
14. Briggs, R.O. and P. Gruenbacher. *EasyWinWin: Managing complexity in requirements negotiation with GSS*: IEEE.
15. Fernández Carazo, A., et al., *Evaluación y clasificación de las técnicas utilizadas por las organizaciones, en las últimas décadas, para seleccionar proyectos*. REVISTA DE MÉTODOS CUANTITATIVOS PARA LA ECONOMÍA Y LA EMPRESA, 2008: p. 67–115.