

Modelo para medir la calidad de activos de Ingeniería de Requisitos

Model for measuring the asset quality of Engineering Requirements

Kimberly Sánchez Holguín¹, Omar Hurtado Jara² y Gonzalo Génova Fuster³

Recibido: 20 de mayo de 2016
Aceptado: 22 de junio de 2016

Resumen

En este artículo se presenta un modelo de calidad para medir y asegurar la calidad de los activos requeridos de la ingeniería, como los requisitos individuales, patrones de requisitos y exigencias estructuras organizativas. El modelo se basa en una serie de factores o cualidades que creemos que un activo de calidad debe presentar; cada una de estas cualidades serán evaluados de acuerdo con un conjunto de indicadores medibles a través de un sistema de métricas. Los valores obtenidos en ambos, los factores y los indicadores, ayudará a identificar los aspectos prioritarios que deben ser

mejorados en el activo. Sabemos que las problemas principales de software de ingeniería tienen sus raíces precisamente en la ingeniería de requisitos; específicamente en una mala especificación y requisitos de gestión. Por lo tanto, tener requisitos bien definidos mejora el proceso de ingeniería de requerimientos y por lo tanto todo el proceso de desarrollo de software.

Palabras Clave: calidad, métricas, requisitos, patrones.

Abstract

In this article we present a quality model to measure and ensure the quality of the assets of the requirements engineering, such as individual requirements, requirements patterns and requirements organizational structures. The model is based on a number of factors or qualities that we think a quality asset must present; each of these qualities will be evaluated according to a set of measurable indicators through a system of metrics. The values obtained in both, the factors and the indicators, will help to identify the priority aspects which

should be improved in the asset. We know the main problems software engineering faces have their root precisely in requirements engineering; specifically in a poor specification and requirements management. Therefore, having well defined requirements improves the requirements engineering process and consequently whole process of software development.

Keywords: quality, metrics, requirements, standards.

1. Universidad de Piura – Perú kimberly.sanchez@pregrado.udep.edu.pe
2. Universidad de Piura – Perú omar.hurtado@udep.pe
3. Universidad Carlos III de Madrid – España ggenova@inf.uc3m.es

I. INTRODUCCIÓN

La ingeniería de requisitos es la disciplina que se encarga de la captura de necesidades, y especificación y gestión de requisitos de un proyecto de software [22], [17], [24]. Por tanto, la ingeniería de requisitos ejerce un papel primordial en el proceso de desarrollo de software: definir qué se desea producir.

Sin embargo, muchos estudios relacionados a esta área [11], [3], [18], [4] señalan que principalmente una mala especificación y gestión de requisitos son los factores principales de los problemas en la producción de software, ninguna otra parte del trabajo afecta tanto al software como la mala definición de los requisitos [3].

En este sentido, sabiendo que la ingeniería de requisitos es un factor primario para el éxito o fracaso de un proyecto de software, surgen muchos estándares, herramientas, métodos y técnicas con la intención de dar solución al problema: mejorar la calidad de la especificación de requisitos.

Hablar de calidad es entrar a un concepto complejo, existen múltiples interpretaciones del término de diferentes autores e instituciones [6], [5], [14], [13]. En la ingeniería de requisitos, como el producto son los requisitos, la calidad dependerá de qué tan bien la especificación de requisitos, refleje lo que realmente necesita el cliente (El cliente del software propiamente dicho y los ingenieros de software).

Esta necesidad de un buen entendimiento de la especificación de los requisitos, ha provocado que, en gran medida, los requisitos sean expresados en lenguaje natural [18], [15], [23].

Al respecto han surgido muchos estándares [7], [12], [20] y autores [2], [20], [21] que proponen un conjunto de características deseables de los requisitos. Asimismo tenemos autores con propuestas para la mejora de la calidad de requisitos desde el punto de vista lingüístico [8], [9], y también los que brindan un soporte automático para el proceso de calidad de los requisitos [23], [16], [19]. Con apoyo en ello, en el trabajo de investigación realizado, aportamos una solución al problema planteado de la ingeniería de requisitos: el perfeccionamiento de un modelo de medida de la calidad de los activos del ámbito de la ingeniería de requisitos. Todo esto con el fin de mejorar la especificación y, por este medio, también la gestión de los requisitos. De este modo también mejoraremos el proceso de la ingeniería de requisitos y por consiguiente todo el proceso de desarrollo de software, aumentando la probabilidad de éxito de un proyecto.

El modelo propone evaluar la calidad de tres tipos de activos del ámbito de la ingeniería de requisitos. Estos tipos de activos seleccionados para alcanzar los fines generales de la investigación, son los siguientes:

- Representación individual de requisitos. Un requisito específico descrito con un conjunto de atributos descriptivos, relaciones con otros requisitos y relaciones con otros elementos.

- Patrón de requisitos. Un conjunto de requisitos relacionados que representan una solución a un problema recurrente del ámbito de la ingeniería de requisitos.

- Patrón de estructuras organizativas. Una estructura organizativa constituida por tipos o clasificaciones de requisitos que representan una manera de organizar los requisitos para una mejor gestión de los mismos.

El resto de este texto está organizado de la siguiente manera. En la sección II describimos el modelo propuesto, cada uno de los elementos que forman parte de él, así como la interacción entre ellos. Posteriormente en la sección III se detalla el proceso a seguir para conseguir una calificación final del activo. Finalmente en la sección IV presentamos las conclusiones a las que se llegó respecto al tema y los trabajos futuros que este conlleva.

II. MODELO DE CALIDAD

En el modelo propuesto planteamos que para aplicar el control de calidad de los activos, se debe definir una combinación de cualidades deseables a las que llamamos “factores de calidad”. Estos factores son cuantificables mediante un sistema de métricas formado por elementos definidos para el modelo. Estos elementos permiten una medición cuantitativa y cualitativa efectiva y práctica para cada factor de cada activo. El modelo se muestra en la Fig. 1.

Se considera un activo como un subproducto resultado del proceso de la ingeniería de requisitos. El modelo se ha definido para trabajar con tres tipos de activos: requisito individual, patrón de requisitos y patrón de estructuras organizativas; los mismos que ya fueron definidos en la introducción. La clase “activo” representa al activo específico que se pretende evaluar. Mientras que la clase “tipo de activo” representa a los 3 tipos de activos definidos. A continuación en las siguientes subsecciones describimos los elementos que conforman el modelo de calidad que están basados en trabajos previos de los propios autores [10].

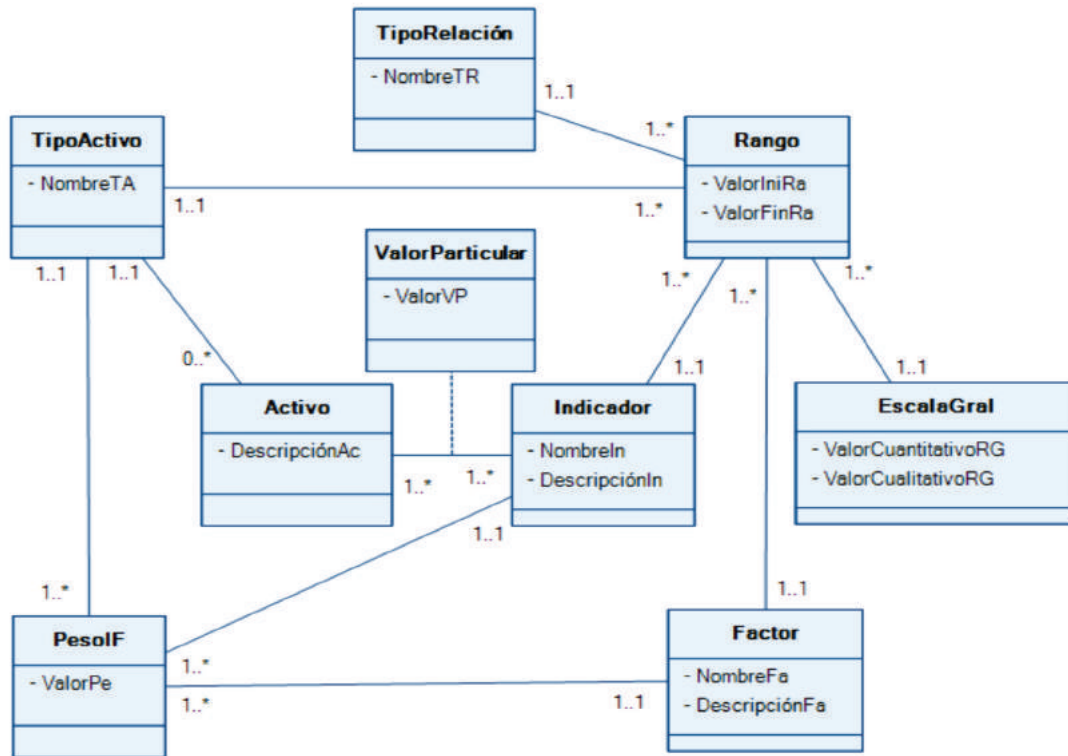


Figura 1: Modelo de calidad de activos de ingeniería de requisitos

A. Factores de calidad

Definimos el “factor de calidad” como una cualidad apreciable en un requisito individual, patrón de requisitos o patrón de estructura organizativa. Esta cualidad permite valorar como buena o mala la calidad de los activos software. Para obtener la lista de cualidades principalmente se tomó en cuenta la información relacionada, la naturaleza de los factores o cualidades deseables, la naturaleza de los activos de requisitos y el punto de vista de los interesados en la calidad de los activos de requisitos. La lista de factores determinados se describe a continuación:

- Validabilidad. Es la cualidad por la cual el activo se corresponde con los deseos o necesidades del cliente.
- Verificabilidad. Determina si el activo de requisitos tiene la capacidad de ser comprobado de manera objetiva una vez que sea implementado.
- Modificabilidad. Determina si el activo tiene las condiciones que permitan su fácil mantenimiento ante los frecuentes cambios que puede sufrir durante todo el proceso de desarrollo de software, así como los futuros cambios del software ya realizado.
- Integridad. También encontrada en la información como “completitud”. Los elementos de un activo deben englobar todos los aspectos básicos y esenciales del problema o situación que se está tratando.
- Consistencia. También conocida como “concordancia”. Está referido a minimizar la posibilidad de encontrar contradicciones entre los elementos de un activo. Asimismo, relacionamos con esta cualidad el evitar las redundancias y solapamientos.
- Inambigüedad. La especificación de un activo no debe dar lugar a más de una interpretación.
- Atomicidad. Cada activo debe ser visto como una unidad coherente sin que se mezcle con otros activos, además describe que el activo responde a una situación o problema específico, aunque este puede presentarse en muchos entornos distintos.
- Abstracción. Especificar claramente lo que el producto software debe hacer, sin necesidad de establecer cómo realizarlo. Se debe evitar el exceso de detalles técnicos en la especificación del activo.
- Comprensibilidad. Un activo de requisitos es comprensible en la medida que su especificación es entendida sin dificultad. Se debe buscar que esté bien redactado, especificando claramente su alcance.
- Precisión. También conocida como “concisión” o “exactitud”. Esta cualidad busca que la especificación de un activo este bien orientada. El alcance de un activo debe ser concreto (bien detallado) y exacto. Esta cualidad puede confundirse con la ambigüedad, pero el activo podría tener una sola interpretación pero no tener la orientación exacta que se requiere.
- Frecuencia. Un activo será frecuente cuando el problema específico al que ha dado solución se presenta muchas veces y en múltiples entornos distintos.
- Coherencia. Todos los elementos de un activo deben tener un orden lógico. Así, un activo es coherente cuando todos sus elementos están alineados a un mismo fin común.

- **Correctitud.** Por definición un activo será correcto cuando esté libre de errores, específicamente los referidos a las reglas que rigen el lenguaje que se utiliza.
- **Sencillez.** Para calificar un activo como sencillo se debe considerar el tipo de lenguaje que se utiliza; sobre todo en la descripción del mismo. Mientras menos palabras técnicas o específicas estén presentes en la especificación de un activo, mayor será la posibilidad de que el lector lo entienda.
- **Adaptabilidad.** Capacidad de un elemento para adecuarse a un nuevo entorno en el que se desempeñará sin cambiar su esencia.
- **Simplicidad.** Mientras menos partes y mientras menos complejas sean las relaciones que existan entre una parte y otra, más simple es el activo.
- **Organización.** Este factor hace referencia al orden de los elementos dentro de la estructura del activo. Un activo para reuso debe entenderse fácilmente y el orden ayuda en gran medida, especialmente en un patrón de estructura organizativa.

B. Indicadores de calidad

Definimos un indicador de calidad como una métrica formalizada para cuantificar las cualidades deseables que se identificaron en la sección anterior. Las cualidades son apreciadas de manera subjetiva según lo que el observador esté buscando o el rol que desempeñe dentro del proyecto. Esta subjetividad no implica que sean arbitrarios, sino difícilmente cuantificables. En este sentido, necesitamos establecer un sistema de medida efectivo para evaluar las cualidades.

Los indicadores determinados en este modelo no son arbitrarios ni suficientes para fijar si una cualidad está presente o no en un activo; sino que serán un punto de referencia para los ingenieros. Lo que se busca es aproximar y facilitar el trabajo. Siempre será necesario el juicio de quien utilice las métricas para llegar a alguna conclusión final.

Como los atributos de los activos típicamente son llenados en lenguaje natural [18], para el modelo principalmente escogimos algunos indicadores basados en el análisis de textos. Pero también podemos encontrar indicadores basados en la estructura misma de los activos, fundamentalmente para los patrones (componentes, relaciones, etc.). Para una mejor descripción de los indicadores desarrollados los clasificamos en cinco grupos principales, según su naturaleza [10].

1) Indicadores morfológicos:

a) **Tamaño:** Se enfoca básicamente en las dimensiones del activo a analizar.

- **Número de palabras:** como el sistema de medida es un punto de referencia, escogimos el número

de palabras para determinar el tamaño porque dará una idea más general de la dimensión del activo, y es una forma de medida más útil en orientación al objetivo buscado.

- **Número de elementos de la solución:** este indicador está orientado a los patrones, tanto de requisitos como de estructuras organizativas. Un patrón presenta un conjunto de componentes que forman una solución a un problema recurrente, que puede ser reusado muchas veces.

b) **Legibilidad lingüística:** Adecuada redacción de un texto, de tal forma que sea fácil de leer. En general mientras más simples o reducidos sean los elementos del texto de especificación del activo a analizar, más legible será ya que implicará menos esfuerzo al entenderlo.

- **Promedio de sílabas por palabra:** mientras más corta es una palabra es más fácil de entender en el conocimiento general.

• **Promedio de palabras por oración:** así como es más legible una palabra de menor cantidad de sílabas, una frase u oración será más legible en cuánto posea menos palabras, sobre todo en los atributos de un patrón en los que se busca precisión.

• **Promedio de frases u oraciones por párrafo:** Un párrafo debe representar una idea, pero esta idea debe estar sustentada por algunas premisas, y si es el caso, mostrar algunas consecuencias. Cada uno de estos elementos los representamos por medio de una frase u oración. En este sentido, si excedemos el número de premisas o consecuencias, aumenta el riesgo de hacer menos entendible a la idea principal.

c) **Puntuación:** El excesivo uso de los signos de puntuación o la ausencia de los mismos puede hacer un texto más difícil de leer y de entender.

• **Promedio de signos de puntuación por frase u oración.** Como está descrito, este indicador hace referencia al uso óptimo de los signos de puntuación para hacer más entendible una frase y a la vez evitar una diferente interpretación. Estos precisan usarse en la medida justa y cuando sean estrictamente necesarios.

d) Acrónimos y abreviaturas:

• **Número de acrónimos o abreviaturas.** Existen acrónimos o abreviaturas de uso frecuente o que pueden estar bien definidas en el glosario del documento, pero el abuso de estos elementos aumenta la posibilidad de perjudicar la calidad de los activos analizados al asumir su conocimiento por parte del lector.

2) Indicadores léxicos:

a) Términos conectivos:

• **Número de conjunciones copulativo-disyuntivas:** Cuando utilizamos conjunciones lo que estamos haciendo es "mezclar" dos oraciones en una sola, lo cual puede crear problemas de claridad y legibilidad al mo-

mento de analizar un texto y entender qué es lo que se debe hacer en sí. Sin embargo, el uso de estos términos pueden ser perfectamente legítimo cuando la intención es especificar precisamente una condición lógica [15], [1], por lo que esta medida debe ser definido y manejado con cuidado.

- Número de términos negativos: Pueden aumentar el riesgo de generar inconsistencias lógicas en el texto. Estos términos pueden ser “no”, “ni”, “nunca”, “nada”, “ninguna” etc.
- Número de términos de flujo de control: son términos que pueden sobrepasar los límites de especificación que deben mostrar los requisitos (el “qué” debe hacer el software y no el “cómo”). Entre estos términos tenemos por ejemplo: “mientras”, “cuando”, “si... entonces”, etc.
- Número de términos anafóricos: hace referencia a los términos capaces de sustituir a otros términos. En este grupo podemos encontrar a los pronombres personales (“él”, “ello”), pronombres relativos (“que”, “donde”), los pronombres demostrativos (“éste”, “ése”, “aquél”), etc.

b) Términos imprecisos: Un defecto importante que debe ser eliminado es el relacionado con los términos imprecisos que incrementa la ambigüedad de la especificación [1], [23]. Los principales son:

- Número de términos imprecisos de calidad: “bueno”, “malo”, “adecuado”, etc.
- Número de términos imprecisos de cantidad: “bastante”, “poco”, “algo”, etc.
- Número de términos imprecisos de frecuencia: “casi siempre”, “usualmente”, etc.
- Número de términos imprecisos de enumeración: “no limitado a”, “varios”, etc.
- Número de términos imprecisos de probabilidad: “posiblemente”, “probablemente”, etc.
- Número de términos imprecisos de usabilidad: “adaptable”, “fácil”, “familiar”, etc.

3) Indicadores analíticos:

- Número de errores de ortografía y gramática: las palabras correctamente escritas y conjugadas influyen en la inteligibilidad del texto.
- Número de términos ambiguos: son términos que admiten dos o más significados distintos (“Luna, satélite o apuntar”) Pues al entenderse de varios modos generan dudas razonables.
- Número de formas verbales condicionales: Es recomendable utilizar formas verbales simples que disminuyan la posibilidad de dudas en el lector. Especialmente se recomienda evitar las formas verbales condicionales (“abriría”, “debería”, etc.), pues estas hablan de casos hipotéticos, mientras lo que se busca en un activo es precisión.
- Número de términos de dominio: Es el conjunto de palabras o expresiones propias de un entorno específico. Conviene utilizar menos cantidad de términos del dominio del proyecto, pues así facilita la adecuación del activo a un entorno distinto a aquel en el que se desarrolló.

4) Indicadores relacionales:

Muestran las conexiones del activo con otros elementos, ya sean estructuras, páginas web, artículos, especificaciones, etc. o incluso otros activos. También toma en cuenta las relaciones existentes entre los elementos internos del propio activo si es el caso.

- Número de relaciones con otros elementos: conexiones que pueden existir entre el activo y otro tipo de elemento de su entorno u otro activo de su misma naturaleza.
- Número de dependencias entre elementos del activo: Los elementos los patrones de requisitos y estructuras organizativas, están relacionados unos con otros y se complementan entre sí.
- Número de solapamientos: esta relación se manifiesta entre activos que tienen partes funcionales o algunas características estructurales comunes entre sí.

5) Indicadores de desarrollo

Manifiestan cómo ha evolucionado el activo.

- Número de versiones: las versiones de un activo surgen cuando se le hacen pequeñas variantes para adaptarlo a un grupo de situaciones más específicas, sin cambiar el núcleo del activo ni su naturaleza.
- Número de éxitos: cuántas veces ha sido utilizado el activo de forma exitosa.
- Número de niveles de anidamiento: cuanto más complejo es un patrón se hace más necesario utilizar más niveles de anidamiento, sin embargo, no se debe llegar a la exageración pues resultaría contraproducente.

Si bien los indicadores considerados para el modelo han sido explicados en la sección “B” de manera general, hay que tener en cuenta que cada tipo de activo se ve afectado por un grupo determinado de ellos, es decir, no todos los indicadores afectan a todos los tipos de activos. Por ejemplo, en la tabla I se muestra los factores establecidos para evaluar la calidad de un patrón de requisitos, y los indicadores que afectan a cada uno de ellos.

Tabla I: Relación de factores e indicadores que permiten evaluar la calidad de los patrones de requisitos

	Atomidad	Precisión	Abstracción	Inambigüedad	Comprensibilidad	Corrección	Verificabilidad	Modificabilidad	Integridad	Frecuencia	Sencillez	Coherencia	Consistencia	Simplificidad	Adaptabilidad
Número de elementos de la solución.	x							x	x					x	x
Número de palabras.	x								x						x
Promedio de palabras por oración.					x						x				
Promedio de frases por párrafo.	x				x										x
Número de abreviaturas o acrónimos.					x						x				
Promedio de signos de puntuación por frase.				x	x										
Número de términos conectivos.	x	x		x	x				x						
Número de términos imprecisos.		x		x			x								
Número de términos de diseño.			x					x			x				x
Número de errores de ortografía y gramática.				x	x	x					x				
Número de términos ambiguos.		x		x	x										
Número de formas verbales condicionales y compuestas.		x									x				
Número de términos de dominio.		x	x								x				x
Número de relaciones con otros elementos.	x							x							x
Número de dependencias entre elementos del activo.								x					x	x	x
Número de versiones.									x						
Número de éxitos.										x		x			

C. Peso del indicador

Ya se han definido los factores, indicadores y tipos de activos. En base a estos elementos se define el peso. El "PesoIF" representa la ponderación o importancia que tiene un indicador respecto a otro dentro del grupo de indicadores que permite evaluar a un "factor" en función a un "tipo de activo". Por ejemplo: para evaluar la "inambigüedad" de un "patrón de requisitos", asumamos que lo hacemos por medio de los siguientes indicadores: "Número de términos imprecisos (NTI)", "Número de signos de puntuación por frase" (NPF) y "Número de fallos ortográficos o gramaticales (NFO)". Como la evaluación es respecto a la "inambigüedad" del patrón de requisitos, la importancia de los indicadores debe ser definida sobre la base del indicador que afecta más al nivel de inambigüedad del texto. En este caso determinamos que el indicador que menos afecta, de los tres definidos es el NTI, al que le ponemos un peso 2, mientras los que más afectan son: NPF y NFO. Y consideramos que para la inambigüedad de un texto el NFO es más importante que el NPF, por lo que colocamos de 4 y 3 respectivamente.

D. Tipo de relación factor-indicador

La clase "tipo de relación" representa al tipo de relación que tiene el indicador respecto al factor del activo respectivo. La forma en que cada factor de un activo se relaciona con cada uno de sus indicadores varía; por ejemplo, en algunos casos, una medida alta del indicador será mejor para el factor, mientras que en otros casos una medida baja del indicador será lo deseable. Dependiendo cómo se correspondan los factores e indicadores, los tipos de relación pueden ser de cuatro tipos principales:

- Tipo de relación creciente. Cuanto más alto sea la medida del indicador la calidad respecto al factor es mejor.
- Tipo de relación decreciente: Cuanto más alto sea la medida del indicador, la calidad respecto al factor es peor.
- Tipo de relación convexa. Ante una medida intermedia del indicador, el factor se califica como alto y para las medidas extremas (pequeñas o grandes) el factor se califica como bajo.
- Tipo de relación cóncava. Ante una medida intermedia del indicador, el factor se califica como bajo y para las medidas extremas (pequeñas o grandes) el factor se califica como alto.

E. Escalas de las métricas

Al valor específico de la medida de un indicador en un activo concreto le denominamos "valor particular". La clase "valor particular" es la clase de asociación entre las clases "activo" y la clase "indicador". Esta clase representa el valor medido que tiene el activo al momento que es evaluado respecto a un indicador determinado, por ejemplo al medir el "promedio de palabras por oración" se obtiene el valor de 16.58 palabras.

Ahora bien, al manejar diferentes indicadores para un mismo factor, surge un problema: el valor particular de un indicador tiene unidad de medición distinta al que tiene otro indicador, y por lo tanto, no son comparables en un mismo sistema de medición. Por ejemplo, para el factor "adaptabilidad" de un "patrón" se establece un indicador "número de palabras (NP)" y un indicador "número de términos de diseño (NTD)". Mientras el NP se maneja dentro de una unidad de medida que va de 1 a 500, el indicador NTD se maneja dentro de una unidad de medida que va de 0 a 5. Al analizar un activo determinado, resulta que el indicador NP marca 200 y el indicador NTD marca 5. A partir de aquí surgen las preguntas: ¿puedo promediar estos valores?, ¿qué me dicen respecto al factor?, ¿los valores son buenos o malos? Respecto a la pregunta si ¿podemos promediar estos valores?, definitivamente no. Sin embargo, para salvar este problema, podemos definir una "escala general" que permita unificar las mediciones de los distintos indicadores.

La clase "escala general" representa a una unidad de medida genérica en la cual confluyen las unidades de los distintos indicadores en forma unificada. Para nuestro modelo de calidad, asumimos que las unidades de medida de todos los indicadores, tendrán rangos en donde la medida se considere "bueno",

"medio" o "malo" respecto al factor de un activo determinado. Y si a nuestra escala cualitativa le asignamos valores numéricos, de modo que "bueno" equivale a un 1, "medio" a un 2 y "malo" a un 3, entonces tendremos a la vez una escala cuantitativa que nos permita un mejor manejo en el proceso de la evaluación de calidad de los activos de requisitos.

1) Rangos de las métricas

Con la "escala general" resolvemos el problema de la unificación de los sistemas de medida de los distintos indicadores. Pero aún quedaron pendientes las preguntas ante un valor en las unidades del sistema de medida del indicador particular: ¿qué nos dicen respecto al valor?, ¿el valor es bueno, medio o malo? Para solucionar este problema, planteamos que a un rango de valores de un indicador en su sistema particular de medida, le corresponde una única escala general dependiendo de la relación entre el indicador y el factor respectivo de un tipo de activo concreto.

La clase "rango". Es una clase intermedia entre las clases factor, indicador, tipo activo, escala general y tipo relación. Representa un intervalo de valores del sistema de medida particular de un indicador respecto a un factor de un tipo de activo. Para aclarar este punto, nos referimos al ejemplo ya citado en la sección de la escala general. En este ejemplo, referimos a uno de los rangos del indicador NTD respecto al factor "adaptabilidad" del tipo de activo "patrón de requisitos", específicamente el rango "de 0 a 3", que representa el valor "bueno" de la escala general, y que contiene los valores menores de medida, en respuesta al tipo de relación decreciente que presenta el factor con el indicador. En la tabla II se muestra los valores mencionados en el ejemplo.

Tabla II: Rangos del indicador "Número de términos de diseño" del factor adaptabilidad del patrón de requisitos

Número de términos de diseño		
Adaptabilidad	Escala general	
Decreciente		
0 - 3	Bueno	1
4 - 4	Regular	2
≥ 5	Malo	3

III. PONDERACIÓN Y EVALUACIÓN FINAL

Una vez definido los factores y sus indicadores, de acuerdo a lo explicado en la sección del modelo de calidad, necesitaremos los siguientes elementos para realizar la evaluación en cuestión: el tipo de relación (creciente, decreciente, convexa, cóncava), los rangos de valores, y el peso relativo del indicador.

Cada uno de esos elementos fue establecido en esta investigación para cada uno de los tipos de activos. Es importante recordar que los rangos establecidos para un mismo indicador son diferentes para cada factor al que afectan. Como ejemplo, la tabla II especifica los rangos de valores del indicador "Número de abreviaturas y acrónimos" por cada factor al que mide dentro del activo "Patrón de requisitos", además del tipo de relación factor-indicador a la que se debe la ubicación de los rangos respecto de la escala general.

Tabla III: Rangos del indicador "Número de términos conectivos" por factor del patrón de requisitos

Número de términos conectivos						
Atomocidad	Precisión	Inambigüedad	Comprensibilidad	Integridad	Escala General	
*Dec.	Dec.	Dec.	Dec.	*Crec.		
0 - 10	0 - 10	0 - 8	0 - 8	> 12	Bueno	1
11 - 15	11 - 15	9 - 12	9 - 12	6 - 12	Regular	2
> 15	> 15	> 12	> 12	0 - 5	Malo	3

*Dec: Decreciente

*Crec: Creciente

Entonces, haremos uso de los rangos para dar un valor cualitativo general al indicador a partir del valor particular que se obtenga de medir el indicador en el patrón, y obtener la calificación total del factor. Por ejemplo, evaluaremos el factor comprensibilidad de un patrón de requisitos al que denominaremos "Gestión de inventarios".

En la tabla IV se muestra los indicadores que miden la comprensibilidad de un patrón de requisitos, en la segunda columna el peso relativo asignado a cada indicador para la comprensibilidad, y en la tercera el valor cuantitativo de la escala general que le corresponde a cada uno, esta valor fue obtenido después de medir el indicador y ubicar la medida dentro de sus rangos establecidos.

Tabla IV: Resultados del análisis del factor comprensibilidad del patrón "gestión de inventarios" de requisitos

Indicador	Comprensibilidad	
	Peso	Valor de escala
Promedio de palabras por oración	3	2
Promedio de frases por párrafo	1	2
Número de abreviaturas o acrónimos	3	1
Número de signos de puntuación por frase	4	1
Número de términos conectivos	1	3
Número de errores de ortografía y gramática.	3	3
Número de términos ambiguos	2	1
	17	

Como ejemplo podemos mencionar que, al analizar el patrón se contaron 20 términos conectivos (NTC) y 1 término ambiguo (NTA). De acuerdo a los rangos de valores de esos indicadores para la comprensibilidad, corresponde una calificación de "Malo" para el indicador NTC y "Bueno" para el indicador NTA.

Una calificación de cada indicador aún no permite apreciar de forma clara si el factor dentro del patrón es adecuado o no. Por un lado el "número de términos conectivos" nos dice que la comprensibilidad del patrón es mala, mientras que el "número de tér-

minos ambiguos" la califica como buena. Entonces, ¿qué tan bueno es el patrón respecto a su comprensibilidad?

La pregunta anterior la responderemos mediante el promedio ponderado de las calificaciones que asignen sus indicadores. Para hallar la "nota final (NF)" que obtiene el patrón en un factor evaluado, ponderamos los valores cuantitativos que obtuvieron cada uno de los indicadores con sus pesos respectivos, como se muestra en la ecuación (1), donde NF es la nota final.

$$NF = \frac{2 \cdot 3 + 2 \cdot 1 + 1 \cdot 3 + 1 \cdot 4 + 3 \cdot 1 + 3 \cdot 3 + 1 \cdot 2}{17} = 1.74 \quad (1)$$

Número de términos conectivos						
Atomocidad	Precisión	Inambigüedad	Comprensibilidad	Integridad	Escala General	
*Dec.	Dec.	Dec.	Dec.	*Crec.		
0 - 10	0 - 10	0 - 8	0 - 8	> 12	Bueno	1
11 - 15	11 - 15	9 - 12	9 - 12	6 - 12	Regular	2
> 15	> 15	> 12	> 12	0 - 5	Malo	3

Este resultado indica que el promedio ponderado para la comprensibilidad del patrón está entre 1 y 2. En forma cualitativa, en la escala general, el resultado está entre "Regular" y "Bueno".

Se realiza esta operación en todos los factores y así obtener un valor numérico en cada uno de ellos. Una vez obtenido los valores de cada factor, es necesario interpretarlos. Si el valor final es decimal, tendremos que interpretarlo de acuerdo a la proximidad que tenga a los valores de la escala general entre los que se encuentra.

Después de haber evaluado de esta manera todos los factores, así como lo hicimos con la comprensibilidad, lo siguiente por hacer es analizar por qué el patrón necesita revisión respecto a los factores en que obtengamos calificaciones muy cercanas a "Malo" o "Regular". Por ejemplo, en el factor comprensibilidad, si se mira la calificación de los indicadores, encontramos que existen dos indicadores principales calificados como "Malo", estos son el "número de términos conectivos" y el "número de errores de ortografía y gramática", en los que hay que mejorar principalmente, para luego mejorar los que tienen calificación "Regular".

Es muy probable que la mayoría de factores que salgan con calificación "Malo" tengan en común uno o varios indicadores con valor 3, entonces es preciso encargarse primero de ellos para mejorar varios factores a la vez.

IV. CONCLUSIONES Y TRABAJOS FUTUROS

Esta investigación hace una contribución para el mejoramiento cualitativo y cuantitativo de la ingeniería de requisitos. Debido a la importancia de esta fase, también podemos decir que hemos contribuido a la mejora de todo el proceso de desarrollo de software.

El modelo de calidad de requisitos desarrollado consolida coincidencias de los especialistas respecto a las cualidades que deben presentar los requisitos para que cumplan correctamente con su función y garanticen un proyecto exitoso. Todo el esquema conceptual del modelo está pensado para el trabajo con requisitos representados en lenguaje natural. En él se ha definido factores y métricas para evaluar la calidad de los activos software de cada nivel, propuestos en el mo-

delo: requisitos individuales, patrones de requisitos y estructuras organizativas de requisitos. Por la naturaleza misma de los activos de cada nivel, dichos factores y métricas también varían de nivel a nivel.

Podemos concluir que una cualidad no está presente sólo en un tipo de activo, si bien puede darse el caso, la mayoría de ellas precisan estar presentes en varios tipos de activos o en todos ellos. De igual modo los indicadores miden a una o varias cualidades dentro de un mismo tipo de activo evaluado.

El valor obtenido tras medir un indicador puede indicar distintas calificaciones respecto a si el activo es bueno, regular o malo, dependiendo del tipo de activo en el que lo estemos midiendo. Esto debido a que los rangos de las escalas de los indicadores requieren ser más cortos en un tipo de activo y más largos en otro, básicamente por las dimensiones propias del activo.

Los ingenieros de requisitos contarán con una herramienta que haga del documento de requisitos una guía clara de lo que deben diseñar e implementar. Así, ellos tendrán mejores opciones de éxito para implementar el software que contenga las capacidades especificadas descritas en los requisitos; mientras que los clientes sabrán qué esperar acerca del producto que se está desarrollando.

Es importante considerar que los elementos de evaluación y las relaciones entre ellos que contiene el modelo propuesto, pueden ser objeto de discusión entre los especialistas ya que están sujetos a cierto grado de subjetividad, especialmente los rangos de medida de cada indicador. Los resultados obtenidos tras una evaluación deben siempre ser verificados bajo el criterio y análisis de los propios ingenieros. En consecuencia la solución planteada se constituye un instrumento de colaboración que permite notificar a los ingenieros la posibilidad de activos sospechosos. Y es el Ingeniero quien evaluará finalmente al elemento.

Este trabajo deja lugar a una posterior implementación de una herramienta informática que permita la automatización del proceso de evaluación planteado, para hacer de este proceso una técnica de uso eficiente, aparte de efectiva.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Alexander I. and Stevens R., Writing better requirements, Addison-Wesley, Boston, 2002.
- [2] Braude, J. Software Engineering: An Object-Oriented Perspective. John Wiley & Sons, ISBN 0471322083. (2001).
- [3] Brooks, F., No Silver Bullet - Essence and Accidents of Software Engineering, in IEEE Computer, vol. 20(4), ISSN 00189162, 1987, pp. 10–19.
- [4] Christel M. and Kang K., Issues in Requirements Elicitation, Carnegie Mellon Software Engineering Institute, Technical Report, CMU/SEI-92-TR-12, 1992.
- [5] Crosby P., Quality is still free: making quality certain in uncertain times, McGraw-Hill, ISBN 0070145326, 1996.
- [6] Deming E.; Calidad, productividad y competitividad, La salida de la crisis, Ediciones Díaz de Santos, ISBN 8487189229, 1989.
- [7] European Space Agency. ESA Software Engineering & Standardisation: ESA PSS-05-02 Issue 1 Revision 1: Guide to the user requirements definition phase. And, ESA PSS-05-03 Issue 1 Revision 1: Guide to the software requirements definition phase Available: http://www.esa.int/TEC/Software_engineering_and_standardisation/ [last visited on 19-08-2014].
- [8] Fabbrini, F. Fusani, M. Gnesi, S. Lami, G. "The Linguistic Approach to the Natural Language Requirements Quality: Benefit of the Use of an Automatic Tool". Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop, pp. 97-105, 2001.
- [9] Fantechi, A. Gnesi, S. Lami, G. Maccari, A. "Applications of linguistic techniques for use case analysis". Requirements Engineering 8(3):161-170, 2003.
- [10] Génova G., Fuentes J., M. Llorens J., Hurtado O., Moreno V., A framework to measure and improve the quality of textual requirements, in Journal of Requirements Engineering, vol. 18(1), 2013, pp. 25–41.
- [11] Ibañez M. and Rempp H., European User Survey Analysis, Report USV_EUR, version 2.1, ESPITI Project, 1996.
- [12] IEEE Standards Association. IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830-1998. Available: <http://ieeexplore.ieee.org/iel4/5841/15571/00720574.pdf?isnumber=15571&arnumber=720574> [last visited on 22-08-2014].
- [13] International Organization for Standardization, ISO 9000 – Quality management, available: <http://www.iso.org/iso/home.html>, [last visited on 26-11-2014].
- [14] Juran J., Gryna F. and Bingham R., Manual de control de calidad, 2ed, Editorial Reverté, S.A; ISBN 842912652-X, 1990.
- [15] Kasser JE., The first requirements elucidator demonstration (FRED) tool, Syst Eng 7(3), 2004, pp. 243–256.
- [16] Kiyavitskaya, N. Zeni, N. Mich, L. Berry, D.M. "Requirements for tools for ambiguity identification and measurement in natural language requirements specifications". Requirements Engineering 13(3):207-239, 2008.
- [17] Loucopoulos P. and Karakostas V., Systems Requirements Engineering, McGraw-Hill International series in Software Engineering, ISBN 0077078438, 1995.
- [18] Marrero M., Sánchez-Cuadrado S., Fraga A. and Llorens J., Applying Ontologies and Intelligent, in Text Processing in Requirements Reuse, in First Workshop on Knowledge Reuse (KREUSE'08), in 10th International Conference on Software Reuse (ICSR'08), Beijing, China, ISBN 9788469131664, 2008.
- [19] Mich, L. Franch, M. Inverardi, P.L.N.. "Market research for requirements analysis using linguistic tools". Requirements Engineering 9(1):40-56, 2004.
- [20] Ministerio de Administraciones Públicas de España. Métrica Versión 3. Available: http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#U_NV3WPCfnF020139815X [last visited on 19-08-2014].
- [21] Pressman, R. Ingeniería de Software: un Enfoque Práctico. 7ed. McGraw-Hill/Interamericana Editores S.A. ISBN: 9786071503145. (2010).
- [22] Thayer R. and Dorfman M., Software Requirements Engineering, 2ed. IEEE Computer Society Publications, ISBN 0818677384, 1997.
- [23] Wilson WM., Rosenberg LH. and Hyatt LE., Automated Analysis of Requirement Specifications, in: Proceedings of the 19th international conference on software engineering-ICSE'97, Boston, MA, USA, 17–23 May 1997, pp. 161–171.
- [24] Zave P., Classification of Research Efforts in Requirements Engineering, in ACM Computing Surveys (CSUR), vol. 29(4), ISSN: 03600300, 1997, pp. 315-321.