

Modelo de Evaluación de Calidad de Software Basado en Lógica Difusa, Aplicada a Métricas de Usabilidad de Acuerdo con la Norma ISO/IEC 9126

Gustavo Alberto Ruiz, Alejandro Peña, Carlos Arturo Castro,
Angela Alaguna, Luz Mery Areiza y Rafael David Rincón

UNIVERSIDAD DE SAN BUENAVENTURA, Medellín.
gustavo.ruiz@usbmed.edu.co, juan.pena@usbmed.edu.co, carlos.castro@usbmed.edu.co,
carlsaga@epm.net.co, lmareiza@epm.net.co

Recibido para revisión May-2006, aceptado Jun-2006, versión final recibida Jun-2006

Abstract: In this article it is developed and it analyze a computational model based on the concepts of the Fuzzy Logic that it will allow the calculate the value of Usability to final user's of software developments based on the Norma ISO/IECE 9126-1:2.001, and based on the Fuzzy Model of Takagi Sugeno Kang. For this model's development it becomes necessary to keep in mind a series of attributes that they qualify the Usability according to the Norma, like they are: Understanding, Easiness of Learning, Attractiveness, Operability and Conformity. Later on to adjust on the part of the developer final user's application, it will be necessary to fix an Usability value and by means of the use of a genetic algorithm to identify the most sensitive attributes, improving this way the time of development and giving satisfaction to the final user.

1 INTRODUCCIÓN

En este artículo se desarrolla y se analiza un modelo computacional basado en la lógica borrosa, para el cálculo de la usabilidad del software de acuerdo con una serie de criterios de calidad basados en la Norma ISO/IECE 9126-1:2.001. Este modelo surge como respuesta a la necesidad de establecer criterios sólidos de evaluación del software educativo en virtud del incremento en la generación del mismo. Si bien la necesidad surge del campo específico del software educativo, la propuesta pretende ser generalizada a la evaluación del software en general. Para ello se parte de la identificación y evaluación de cinco atributos básicos por parte del usuario final como son: Comprensibilidad, Facilidad de Aprendizaje, Atractividad, Operabilidad y Conformidad.

De los atributos definidos en la Norma, el de usabilidad es especialmente problemático por el alto nivel de subjetividad que presenta, por lo cual se descarta al momento de hacer testing¹ de calidad. Ello conduce a la no aceptación por parte de los usuarios de aplicaciones que cumplen con todos los requerimientos funcionales y no funcionales, debido a las dificultades propias de su uso.

De acuerdo con lo anterior, se desarrolla un modelo de evaluación de calidad del software que permite a

partir de la subjetividad de varios atributos identificados, calcular un valor de usabilidad confiable. Para el desarrollo de este modelo se utilizaron conceptos propios de la lógica borrosa, y en especial el modelo borroso del tipo Takagi Sugeno Kang, el cual a partir de una serie de valores de pertenencia permite el cálculo de un valor para una variable de salida o variable a controlar de manera analítica a partir de la aplicación de criterios estandarizados en la evaluación de la calidad del software.

Una vez que se obtiene el modelo, se hicieron una serie de encuestas a cada uno de los usuarios finales de una determinada aplicación y de acuerdo con un valor de usabilidad preestablecido se estimaron una serie de parámetros que permitieron identificar por parte de ese usuario los atributos en los cuales la empresa desarrolladora se debería ajustar a las normas de la ingeniería del software y de esta manera mejorar la aplicación de usuario final, disminuyendo así el tiempo de desarrollo por parte de la empresa encargada del mismo.

En este artículo se plantea una caracterización general en torno a la evaluación del software y cada uno de los factores que afectan la misma. Posteriormente se hace un análisis de la usabilidad para luego proponer un modelo de tipo borroso que permita ajustar el valor de usabilidad a una serie de factores identificados por parte

¹Proceso a través del cual se obtienen indicadores de calidad con respecto a las características de un producto evaluado. normalmente se realiza por medio de una lista de chequeo.

del usuario final de la aplicación.

2 IDENTIFICACIÓN DE CONCEPTOS ADMINISTRATIVOS DE LA CALIDAD DEL SOFTWARE

La Calidad se determina en función de las características que hacen competitivo un producto al satisfacer las necesidades implícitas y explícitas de los usuarios finales. Esto significa, una responsabilidad hacia el cliente al suministrar las especificaciones correctas. En esencia, se definirá la calidad como: satisfacción de las necesidades y cumplimiento de las especificaciones y fabricación para cumplir con el propósito del cliente. Entre otras los requisitos de calidad hacen referencia al conjunto de requisitos expresados en forma cuantitativa o cualitativa respecto a las características de unos productos. Para hacer posible su realización y examen; los requisitos pueden ser convenidos y basados a partir de las necesidades del mercado interno de una organización o de la sociedad, esto incluye lógicamente requisitos jurídicos y reglamentarios.

Un aspecto fundamental en la calidad, es la de la gestión, la cual contempla la planificación (describe las operaciones que establecen los objetivos y los requisitos de calidad del producto, para el sistema de aseguramiento de la calidad); el control de calidad (con el conjunto de acciones y técnicas operacionales que se usan para cumplir con los requisitos de calidad); el aseguramiento de la calidad (centrado en todas las actividades planificadas y sistemáticas implementadas dentro del sistema de calidad y evidenciadas como necesarias para asegurar los requisitos de calidad) y El sistema de calidad (compuesto por la estructura organizacional, los procedimientos, procesos y recursos necesarios para implementar la administración de la calidad).

En el caso de la calidad de software, está se define como: "La concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente" [Anderson (1992)].

El estándar de la IEEE para la calidad de software [Pressman (2001)] define el término calidad del software como:

- La totalidad de rasgos y características de un producto de software que se refieren a su habilidad para satisfacer necesidades específicas.
- El grado en el cual el software posee una combinación deseada de atributos.
- El grado en el cual un usuario o cliente percibe que el software cumple con sus expectativas.

- Las características del software que determinan el grado en el cual dicho software en uso, cumple con las expectativas del cliente

Para Pressman (2001), se hace necesario medir el software para indicar la calidad del producto, evaluar la productividad de la gente que lo desarrolla, evaluar los beneficios (en términos de productividad y de calidad) derivados del uso de nuevos métodos y herramientas de ingeniería del software, establecer una línea base para estimación, y ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Las métricas para la evaluación del software se pueden catalogar en métricas de productividad (enfocadas al rendimiento) de Calidad (enfocadas al nivel de ajuste a los requisitos explícitos e implícitos del cliente) y Métricas Técnicas (orientadas a características como complejidad, y grado de modularidad, más que en el proceso de desarrollo)

Las métricas mencionadas, puede tener otra clasificación: orientadas al tamaño, la función y a la persona. Las métricas orientadas a la persona facilitan información sobre la manera como la gente desarrolla software, además de la efectividad de las herramientas y métodos utilizados. Éstas son subjetivas pues dependen de las percepciones, opiniones y juicios de las personas, pueden ser cuantitativas o cualitativas, como se puede observar de las de medidas subjetivas utilizadas en un Test de usabilidad típico [IEEE (1983)]:

- Facilidad de uso del producto
- Facilidad de aprender el producto
- Facilidad de hacer una determinada tarea
- Facilidad de instalar el producto
- Facilidad de encontrar información en el manual
- Facilidad de comprender la información
- Utilidad de los ejemplos de ayuda

3 ANÁLISIS DEL CRITERIO DE USABILIDAD

La Norma ISO/IECE 9126-1:2.00 categoriza los atributos de calidad del software en seis características: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, las cuales se subdividen en subcaracterísticas [IEEE (1983)].

- Funcionalidad: La capacidad que tiene un producto de software para proveer funciones que satisfacen necesidades establecidas e implícitas, cuando el software es usado bajo condiciones específicas. Esta compuesta por las siguientes subcaracterísticas: apropiabilidad, exactitud, interoperabilidad y cumplimiento con la funcionalidad.

- Confiabilidad: La capacidad que tiene un producto de software para mantener su nivel de desempeño cuando éste es usado en condiciones específicas. Esta compuesta por las siguientes subcaracterísticas: madurez, tolerancia a fallas, recuperabilidad y cumplimiento con la confiabilidad.
- Usabilidad: La capacidad que tiene un producto de software para ser entendible, aprendido, utilizable y atractivo al usuario cuando éste es usado en condiciones específicas. Esta compuesta por las siguientes subcaracterísticas: comprensibilidad, facilidad de aprendizaje, operabilidad, atractivo y cumplimiento con la usabilidad.
- Eficiencia: La capacidad que tiene un producto de software para proveer el desempeño apropiado relacionado a la cantidad de recursos usados, bajo condiciones determinadas. Esta compuesta por las siguientes subcaracterísticas: comportamiento en el tiempo, utilización de recursos, cumplimiento con la eficiencia.
- Facilidad de Mantenimiento: La capacidad que tiene un producto de software para ser modificado. Modificaciones pueden incluir correcciones, mejoras o adaptación del software a los cambios de entorno, requisitos y especificaciones funcionales. Esta compuesta por las siguientes subcaracterísticas: trazabilidad, facilidad de cambio, estabilidad, facilidad de ensayo, cumplimiento con la facilidad de mantenimiento.
- Portabilidad: La capacidad que tiene un producto de software para ser transferido de un ambiente a otro. Está compuesta por las siguientes subcaracterísticas: adaptabilidad, instalabilidad, coexistencia, reemplazabilidad, cumplimiento con la portabilidad.

4 MODELO BORROSO PARA LA EVALUACIÓN DEL SOFTWARE UTILIZANDO MÉTRICAS DE USABILIDAD

El primer desarrollo del modelo *Takagi Sugeno Kang*, para la evaluación de software utilizando métricas de usabilidad constará de tres partes fundamentales:

1. Definición de las variables lingüísticas del Modelo.
2. Estructura del modelo Takagi Sugeno Kang TSK
3. Ajuste por Evolución y Autoadaptación.

4.1 Definición de las Variables Lingüísticas del Modelo

Las variables lingüísticas que se utilizan en el modelo, estarán determinadas por cada uno de los atributos considerados por la Norma ISO/IEC 9126-1 del 2001, para

métricas de usabilidad como son: Comprensibilidad, Facilidad de Aprendizaje, Atractividad, Operabilidad y Conformidad con la Usabilidad. Cada uno de estos atributos estará definido por una variable lingüística con la siguiente estructura de cualidades como se muestra en la Tabla 1.

Tabla 1: Estructura de cualidades para las variables lingüísticas del modelo

Variable Ling.	$X \leq a_1$	$a_1 < X < b_1$	$b_1 \leq X < c_1$	$X \geq c_1$
Mala	1	$\frac{(b_1 - X)}{(b_1 - a_1)}$	0	0
Regular	0	$\frac{(X - a_1)}{(b_1 - a_1)}$	$\frac{(c_1 - X)}{(c_1 - b_1)}$	0
Buena	0	0	$\frac{(X - b_1)}{(c_1 - b_1)}$	1

Estas cualidades y valores, están de acuerdo con la Norma, la cual define estos valores pero de manera determinística. Para el caso específico del modelo borroso, estos conjuntos fueron definidos mediante un conjunto de cualidades y utilizando un factor de ponderación inicial de acuerdo con la importancia que cada usuario final o empresa le da de cada uno de los atributos de la aplicación. Es de anotar que si una cualidad no es importante para el usuario final o empresa, entonces el intervalo de los conjuntos Malo y Regular disminuye dando una mayor posibilidad a que los conjuntos Bueno y Excelente sean incluidos para el cálculo de la usabilidad.

Los valores asociados con cada una de las variables lingüísticas, estarán determinadas por la representación analítica de las mismas como se muestra en la Tabla 2. Por su parte la Figura 1 muestra una representación gráfica de las ecuaciones definidas en la Tabla 2.

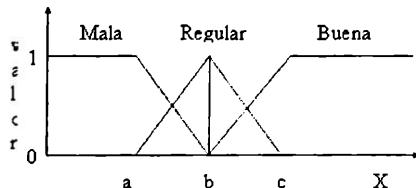


Figura 1: Representación gráfica de las variables lingüísticas

Tabla 2: Definición analítica de las variables lingüísticas

Rango	Variable
0	Mala
45	Regular
60	Buena
75	Excelente

Cada una de las subcaracterísticas consideradas en el modelo poseerá una estructura similar a la mostrada en la Figura 1 y en la Tabla 2.

La conformación de los conjuntos borrosos asociados con cada una de las subcaracterísticas, estarán determinados por una serie de encuestas realizadas a los usuarios potenciales de una aplicación cualquiera, este valor estará dado entre [0, 100%] y quedará definido por el número de puntos que se le asigne a cada encuesta que evalúa cada subcaracterísticas o atributo, por ejemplo para el caso de comprensibilidad la encuesta será de 17 preguntas, cada una con un puntaje de 10 puntos, lo que arrojará un intervalo de valores entre [0, 1700]. De igual manera se hará para cada una de las otras variables del modelo.

De acuerdo con la importancia que cada organización dé a cada una de los atributos de entrada al modelo, los conjuntos borrosos sufrirán un desplazamiento tanto a la derecha, en el caso de mucha importancia, así como a la izquierda en el caso de que se le de poca importancia a la variable.

El movimiento de cada una de los conjuntos borrosos se describe a continuación:

Lista numerada

1. Se asigna un valor de importancia a cada una de las características a evaluar y se calcula el promedio ponderado por porcentaje.
2. Se identifica cual es el mayor y menor peso dentro del conjunto de entrada.
3. Teniendo dichos pesos, es posible definir un intervalo entre el mayor peso (b) y el menor peso (a) y para este se halla el valor medio (m).
4. Se verifica cuales pesos de los definidos en el ítem 1, están por encima y cuales por debajo del valor medio hallado en el ítem 3.

$$m = \left[\frac{(b-a)}{2} \right] + a \quad (1)$$

5. Los conjuntos borrosos en los cuales las subcaracterísticas tengan pesos por encima del valor medio, se desplazan $\left[\frac{(b-m)}{2}\right]$ unidades a la derecha del eje "x" y los que tengan pesos por debajo del valor medio, se desplazan $\left[\frac{(m-a)}{2}\right]$ unidades a la izquierda del eje x.

4.2 Estructura del Modelo Takagi Sugeno Kang para la Obtención del Valor de Usabilidad

El Sistema TSK Fuzzy tiene una estructura que se modela partir de la definición de un conjunto de reglas de la siguiente manera [Nielsen (1990)]:

IF x_1 *is* C_1^l *and* ... *and* x_n *is* C_n^l
THEN $y^l = c_0^l + c_1^l x_1 + \dots + c_n^l x_n$ (2)

Donde, C_i^l son los conjuntos Borrosos, c_i^l son constantes de la combinación lineal de salida, $l = 1, 2, \dots, M$.

Aquí la parte IF de las reglas, es similar a los sistemas Fuzzy convencionales, pero la parte THEN es una combinación lineal de las variables de entrada al sistema. Es de anotar que $x = (x_1, x_2, \dots, x_n)^T \in U \subset R^n$, mientras que la salida del sistema estará dada por $y^l \in V \subset R$ y se describe en la ecuación (3) [Nielsen (1990)]:

$$y^l = \frac{\sum_{i=1}^M y^l w^i}{\sum_{i=1}^M w^i} \quad (3)$$

Donde los pesos de la Combinación Lineal para la salida están dados por la ecuación (4) [Nielsen (1990)]:

$$w^l = \prod_{i=1}^n u_{C_i^l}(x_i) \quad (4)$$

Para el desarrollo del modelo *Fuzzy-Usability*, es necesario inicialmente determinar las variables de entrada y salida a considerar del sistema, para luego realizar una identificación del sistema teniendo en cuenta la siguiente estructura:

$$u = f(c, fa, a, o, co) \quad (5)$$

Donde: u es el valor de usabilidad para el software, c es la comprensibilidad de la aplicación, fa es la facilidad de Aprendizaje, a es la atractividad, o es la operabilidad y co es la conformidad.

Ahora, si la salida depende de los valores de entrada al sistema, se obtiene la función de *TSK Fuzzy*. Específicamente, un *TSK Dinámico* es construido a partir de (5) y de acuerdo con la estructura de las siguientes reglas:

IF c *is* C^p *and* f *is* F^p *and* a *is* A^p
and o *is* O^p *and* co *is* CO^p (6)
THEN $u^p = a_1^p c + a_2^p f + a_3^p a + a_4^p o + a_5^p co$

Donde, C^p, F^p, A^p, O^p, CO^p son los conjuntos difusos de entrada al sistema. $a_1^p, a_2^p, a_3^p, a_4^p, a_5^p$ son las constantes del Sistema. $p = 1, 2, \dots, N$ es el número de reglas del sistema.

4.3 Ajuste por Evolución y Autoadaptación

Para ajustar la aplicación desarrollada, se fija un valor de usabilidad U cualquiera dependiendo de la empresa y luego se evalúa el modelo por cada uno de las personas que van a utilizar la aplicación. A partir de este valor utilizamos un algoritmo de tipo genético (representación binaria) que permite ajustar los valores de $a_1^p, a_2^p, a_3^p, a_4^p, a_5^p$

al valor de usabilidad fijado de la siguiente manera [Takagi y Sugeno (1985)]:

a_1^p	a_2^p	a_3^p	a_4^p	a_5^p
---------	---------	---------	---------	---------

Figura 2. Estructura del individuo a evolucionar

Los valores más altos obtenidos de esta estimación, ayudarán a identificar los factores que más afectan la aplicación desarrollada y se genera un proceso de re-alimentación por parte del usuario final a la empresa desarrolladora, para generar un ajuste significativo del aplicativo al hacer énfasis de manera particular en un atributo en particular y para las necesidades específicas de un usuario final.

5 CONCLUSIONES

1. La definición de cada una de las variables lingüísticas del modelo y de cada uno de los conjuntos borrosos, se ajustaron de manera general a la Norma con algunas modificaciones, dadas específicamente por el carácter cualitativo de los atributos que forman el modelo.
2. El modelo permitió mediante un proceso de evolución, identificar los factores más sensibles dentro de la usabilidad de una aplicación para un usuario final determinado.
3. La identificación de los factores que afectan la usabilidad, permiten a la empresas desarrolladoras centrarse en los aspectos más relevantes de la aplicación, disminuyendo de manera significativa el tiempo de desarrollo en las mejoras que requiere un sistema.

4. La fijación de un valor de usabilidad del sistema y el ajuste de los pesos utilizando técnicas evolutivas, permite ajustar la función de salida del TSK generando de esta manera un modelo ajustable de tipo adaptativo a las necesidades del cliente, lo que permite además mejorar el ciclo de vida de producto y llegar hasta el desarrollo de un sistema PLM (Product LifeCycle Management) para el desarrollo de software.

5. A pesar del componente altamente cualitativo del modelo, los resultados obtenidos se ajustaron a las necesidades de los usuarios finales al estimar a partir de valores cualitativos un valor cuantitativo para la variable de salida por medio del control TSK.

REFERENCIAS

- Anderson, R. (1992), 'Social impacts of computing: Codes of professional ethics', *Social Science Computing Review* 10(2), 453-469.
- IEEE (1983), *IEEE Glossary of Software Engineering Terminology*, Institute of Electrical and Electronics engineers, USA. Product Standard.
- Nielsen, J. (1990), 'Ten usability heuristics', En línea: <http://www.useit.com/papers/heuristic/heuristic_list.html>.
- Pressman, R. (2001), *Ingeniería del Software, un enfoque práctico*, 5a edn, Mexico McGraw Hill.
- Takagi, T. y Sugeno, M. (1985). 'Fuzzy identification of systems and its applications to modeling and control', *IEEE trans Systems, Man and Cybernetics* 15(1), 116-132.

