

Quantitative Methodology to Form Engineers Evaluators of Development Tools for Information Systems

L. S. Vargas Pérez, MSc¹, A.F. Gutiérrez Tornés, PhD², E.M. Felipe Riverón, PhD³, V.A. Vargas-Pérez, MSc⁴, J. Peralta Escobar, MSc¹

¹ I. T. Cd. Madero, laura.silvia.vargas@gmail.com, jperalta3008@gmail.com

²Instituto Tecnológico de Estudios Superiores de Monterrey-CCM, México, agustin.tornes@itesm.mx

³Centro de Investigación en Computación-IPN, México, edgardo@cic.ipn.mx

⁴Universidad Autónoma Tamaulipas, México, vanessa.atenea@gmail.com.

Abstract– To have software tools that facilitate the rapid development of applications and generate information systems is a requirement in a globalized world. Worldwide in the engineers training field it is basic to apply measurements and evaluations to the rapid design tools used to obtain applications faster and generate quality information systems in order to determine the best for this purpose. This paper proposes a methodology for the technical evaluation of the visual environments tools to generate information systems where commercial products are evaluated using as reference basic elements of international standards. With this methodology one can evaluate and select in an effective and easy way, those tools and development platforms best suited to create applications in visual environments, in order to generate information systems with quality and sustainability. It is useful for higher education institutions, organizations, companies and system's end users, among others.

Keywords— software quality, technical evaluation model, visual environment, RAD tools.

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2015.1.1.182>

ISBN: 13 978-0-9822896-8-6

ISSN: 2414-6668

13th LACCEI Annual International Conference: “Engineering Education Facing the Grand Challenges, What Are We Doing?”
July 29-31, 2015, Santo Domingo, Dominican Republic **ISBN:** 13 978-0-9822896-8-6 **ISSN:** 2414-6668
DOI: <http://dx.doi.org/10.18687/LACCEI2015.1.1.182>

Metodología Cuantitativa para formar Ingenieros Evaluadores de Herramientas de Desarrollo de Sistemas de Información

L. S. Vargas-Pérez, MSc¹, A.F. Gutiérrez-Tornés, PhD², E.M. Felipe-Riverón, PhD³,
V.A. Vargas-Pérez, MSc⁴, J. Peralta Escobar, MSc⁵

^{1,5}I. T. Cd. Madero, laura.silvia.vargas@gmail.com, jperalta3008@gmail.com

²Instituto Tecnológico de Estudios Superiores de Monterrey-CCM, México, agustin.tornes@itesm.mx

³Centro de Investigación en Computación-IPN, México, edgardo@cic.ipn.mx

⁴Universidad Autónoma Tamaulipas, México, vanessa.atenea@gmail.com

I. INTRODUCCIÓN

Resumen. En el mundo globalizado se requiere contar con herramientas de software que faciliten el desarrollo de aplicaciones rápidas para generar sistemas de información. En la formación de los ingenieros, a nivel mundial, es básico realizar mediciones y evaluaciones de las herramientas de diseño de aplicaciones rápidas, para generar sistemas de información de calidad, y determinar las mejores para tal fin. En este trabajo se propone una metodología propia para realizar la evaluación técnica de las herramientas para generar sistemas de información, en ambientes visuales, donde productos comerciales puedan ser evaluados, tomando como referencia elementos básicos de estándares internacionales, que son muy importantes en la formación de ingenieros. Con esta metodología se puede evaluar y seleccionar, de manera eficaz y fácil, las herramientas y sus plataformas de desarrollo más adecuados para crear aplicaciones en ambientes visuales, con el fin de generar sistemas de información con calidad y sustentabilidad. Está dirigida a instituciones de educación superior, organizaciones, empresas, usuarios finales, así como a los estudiantes de ingeniería y profesores de educación superior, entre otros.

Palabras clave: calidad de software, modelo de evaluación técnica, ambiente visual, herramientas RAD.

Abstract. To have software tools that facilitate the rapid development of applications and generate information systems is a requirement in a globalized world. Worldwide in the engineers training field it is basic to apply measurements and evaluations to the rapid design tools used to obtain applications faster and generate quality information systems in order to determine the best for this purpose. This paper proposes a methodology for the technical evaluation of the visual environments tools to generate information systems where commercial products are evaluated using as reference basic elements of international standards. With this methodology one can evaluate and select in an effective and easy way, those tools and development platforms best suited to create applications in visual environments, in order to generate information systems with quality and sustainability. It is useful for higher education institutions, organizations, companies and system's end users, among others.

Keywords: software quality, technical evaluation model, visual environment, RAD tools.

Para mejorar cualquier software se requiere medir sus atributos mediante un conjunto de medidas y métricas significativas, utilizadas para proporcionar indicadores que conduzcan a una estrategia de evaluación técnica de la calidad del producto. Al llevar a cabo este proceso mediante un modelo de evaluación de la calidad de software, se cuenta con una metodología para valorar los requerimientos que debe cumplir. Es importante que las medidas de productos de software puedan ser hechas fácilmente y que el resultado de la medición pueda ser interpretado de la misma manera. La forma en la cual las características de calidad han sido definidas no permite que sean medidas directamente, por lo que se requiere establecer métricas que correlacionan estas características en un producto de software [1].

El primer paso en la evaluación de cualquier software debe ser la determinación de las propiedades relevantes de calidad mediante un modelo técnico de evaluación de la calidad de un producto. Un modelo de este tipo identifica los componentes de calidad y sus interrelaciones. Su objetivo es facilitar la evaluación cualitativa y cuantitativa de estos componentes. Esto se hace a través de la obtención de mediciones adecuadas para compararlas con uno o más criterios. Los modelos de evaluación de la calidad de software generalmente representan la totalidad de los elementos a evaluar, clasificados en un árbol de estructura jerárquica donde aparecen las características en su nivel más alto, las subcaracterísticas en el intermedio y los atributos en el más bajo [2], [3].

A. Tipos de medidas

El tipo de medición requerida dependerá del propósito de la evaluación. Si el propósito primario es entender y corregir deficiencias, muchas mediciones pueden ser hechas en el software para monitorear y controlar las mejoras. Para comparar la calidad de un producto con productos alternativos o contra requerimientos, es importante que la especificación de la evaluación se base en un modelo de calidad preciso, métodos de medición, escalas, y rango de niveles para cada métrica [4].

Al nivel internacional surgen, periódicamente, herramientas para facilitar la creación de nuevos sistemas de información para diversas aplicaciones; es imprescindible que se identifiquen estas herramientas como de diseño rápido de aplicaciones, y así puedan ser evaluadas para determinar si cumplen con los requisitos de calidad en uso, establecidos por los proyectos y/o casas de software que las produce. [5], [6], [7], [8].

En la formación de ingenieros de clase mundial, es necesario realizar mediciones y evaluaciones de diversos productos; con respecto al software, es básico realizar mediciones y evaluaciones de las herramientas de diseño y desarrollo de aplicaciones para generar sistemas de información de calidad, y de esta manera determinar cuáles son las mejores para que los programadores las elijan para trabajar. A estas herramientas se les conoce como herramientas RAD (*Rapid Application Development*) las que pueden estar dentro de un IDE (*Integrated Development Environment*-Entorno o Ambiente de Desarrollo Integrado), que es el marco de trabajo preferido de los programadores de aplicaciones para crear sistemas de información.

B. *Objetivo*

Formar ingenieros evaluadores de proyectos, productos y servicios de software, con conocimientos para la evaluación de las herramientas de diseño de aplicaciones rápidas (herramientas RAD) para generar sistemas de información de calidad.

El modelo MECHDAV, Modelo de Evaluación de la Calidad de las Herramientas de Desarrollo de Aplicaciones en Ambientes Visuales (o MECRAD, Modelo de Evaluación de la Calidad de las RAD, nueva versión más completa en inglés), permite hacer un análisis comparativo de diferentes tipos de herramientas para el desarrollo de software de aplicación en ambientes visuales, mediante la utilización de un plan de métricas de las cuales se podrán seleccionar las más adecuadas para medir la calidad de productos de software y analizar cómo éstas permiten caracterizar, evaluar, predecir y mejorar dichos productos.

C. *Estado del arte y trabajos relacionados*

La propuesta a presentar es un modelo y su herramienta de software. Está basado en las normas ISO/IEC9126, ISO/IEC 14598 (Proyecto SQUARE ISO 25000), IEEE 1061 y en otros modelos. Es un modelo estandarizado para la evaluación técnica de las herramientas RAD en ambientes visuales, donde productos comerciales son evaluados tomando como referencia elementos básicos de las normas mencionadas con nuevas adaptaciones, debido a que no se tiene acceso al código fuente de los productos [6], [7], [8], [9].

Desde hace algunos años se han propuesto algunos tipos de modelos muy variados basados en ciertos estándares, tales como el ISO/IEC 9126, entre otros. Estos modelos son muy útiles, pero a su vez muy genéricos, por lo que deben ser adaptados para su utilización para evaluar la calidad de procesos y/o productos de software. Algunos trabajos propuestos se enfocan en la evaluación de procesos de desarrollo de software, como por ejemplo: Moreno [11],

Olsina [12], Piattini [13], Pastor [14]. Sin embargo, el enfoque de este trabajo es diferente, ya que los modelos y estándares que contribuyen con él, tales como la ISO/IEC 9126, ISO/IEC 14598, ISO/IEC25000, IEEE1061, MECA [2], MACS [3], SUMI [10], se toman como referencia para MECRAD en lo siguiente:

- De los modelos MECA y MACS (etapa: proceso “Prueba de aceptación”), los cuales enfatizan 7 Características, 22 Subcaracterísticas, 42 Atributos y no se definen métricas; de ellas se tomaron como referencia para MECRAD 5 Características, 11 Subcaracterísticas y 10 Atributos.
- Del modelo ISO/IEC 9126, se readaptan las métricas externas y de calidad en uso las cuales enfatizan 6 Características, 21 Subcaracterísticas, donde le falta definir con mayor precisión los atributos y desarrolla 125 métricas; de lo anterior se toman como referencia para MECHDAV (MECRAD) 5 Características, 11 Subcaracterísticas y 14 Métricas.
- De la norma ISO/IEC 14598 (Etapas Procesos para Compradores y Evaluadores), solo define una sola Característica, 3 Subcaracterísticas y 3 Atributos, pero no define métricas; se tomaron como referencia sólo 3 Subcaracterísticas.
- Del SUMI (Inventario de Medidas de Usabilidad del Software), se define una sola Característica, 1 Subcaracterística y 48 Atributos; se tomó como referencia la Subcaracterística mencionada y 11 Atributos.

II. METODOLOGÍA

Se genera una metodología propia (con modelos, procesos, técnicas y herramientas), que permite hacer cuadros comparativos para realizar la evaluación técnica de las herramientas RAD y para generar sistemas de información diversos en ambientes visuales; con ella, los productos comerciales IDE pueden evaluarse tomando como referencia los elementos básicos de los estándares internacionales, como los ya mencionados, todos ellos muy importantes en la formación de los ingenieros.

La arquitectura del modelo propuesto se basa en la fusión de las normas mencionadas, ISO/IEC 9126, 14598, proyecto SQUARE ISO/IEC 25000 y IEEE 1061, de las cuales se adopta una parte que se utiliza para conformar la arquitectura de este modelo. Es necesario recalcar que los productos de software para cuya evaluación técnica se diseña el modelo están ya en etapa de operación. Por ser productos comerciales terminados, no se tiene disponible la información concerniente a su desarrollo y a su código fuente, por lo que no se toman en cuenta las métricas internas.

A. *Proceso de evaluación*

Para evaluar la calidad del software, primero se establece la evaluación de los requerimientos de calidad y de ahí es que se especifica, diseña y ejecuta la evaluación. Las actividades de evaluación a realizar dentro del modelo propuesto son las que se indican en un proceso que comprende cinco actividades (Figura 1). Es por eso que tendrán una importancia particular aquellos atributos relacionados con la calidad en uso, que es el efecto

combinado de la medición de las características de calidad desde el punto de vista del usuario.

La relación de la calidad en uso con las otras características de calidad del software depende del tipo de usuario. Por ejemplo, el *Usuario final*, para quien la calidad en uso es principalmente el resultado de la evaluación de la funcionalidad, la confiabilidad, la usabilidad y la eficiencia, junto con una nueva característica o requisito de calidad: la reusabilidad [ISO/IEC25000, 2005] [9].

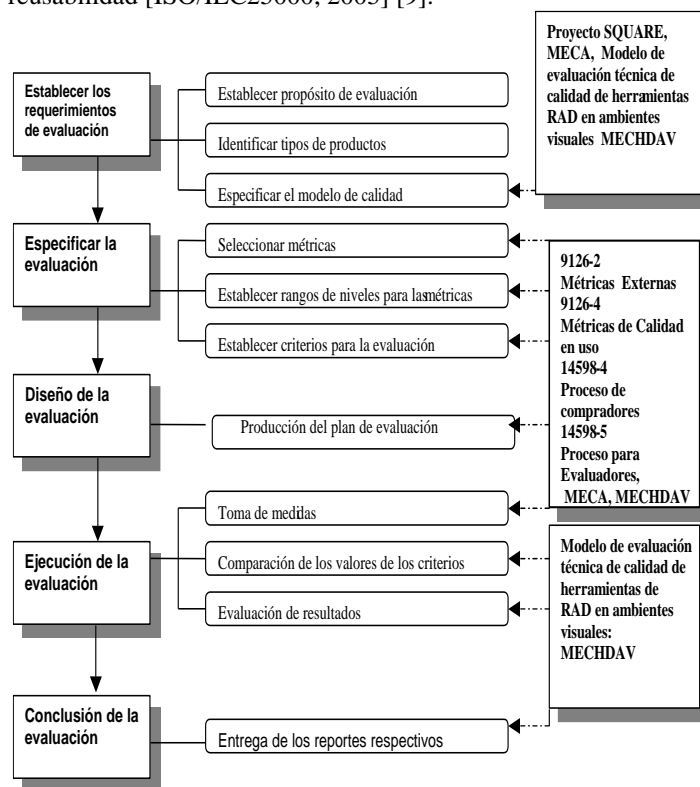


Fig. 1. Proceso de evaluación para el modelo propuesto MECHDAV (MECRAD) [15].

B. Compactación del modelo MECHDAV (MECRAD)

Para el modelo MECHDAV se proponen las siguientes cualidades específicas del software (características), así como sus respectivas subcaracterísticas, atributos y métricas para la evaluación sugerida para usuarios expertos (modelo completo), recomendado para los profesores y alumnos de semestres avanzados, de ingeniería y posgrado que desarrollan aplicaciones de software, y para usuarios novatos (submódulo del principiante), recomendado para los alumnos de los primeros semestres de ingeniería dentro de sus materias de desarrollo de software. En la Figura 2 se muestra el modelo MECHDAV completo recomendado para usuarios expertos, y en la Figura 3 se muestra un subconjunto de este modelo para evaluar herramientas RAD, hechas por usuarios inexpertos.

En este modelo se propone un conjunto de métricas básicas, las cuales dan la medida o pueden ser aplicadas para representar las características de calidad del tipo de software propuesto a evaluar. Uno de los propósitos de la aplicación de este modelo de evaluación técnica, es la de poder proporcionar un medio de comparación serio entre diversos

productos de software dentro de un ambiente visual para cualquier tipo de usuario (sea experto o principiante).

C. Definición de las métricas y de la escala evaluativa

Las características cuantificables deben ser medidas, cuantitativamente, mediante métricas. El resultado, el valor medido, puede ser mapeado sobre una escala. Este valor no muestra en sí mismo el nivel de satisfacción de los requerimientos. Para este propósito la escala tendría que ser dividida en rangos correspondientes a los diferentes grados de satisfacción.

Características / Subcaracterísticas / Atributo / Métrica

- 1.1.1.1 Funcionalidad / **Compleitud** / Contenido total / métrica
- 1.2.1.1 Funcionalidad / **Consistencia** / Uniformidad del formato de los elementos funcionales / métrica
- 1.2.2.1 Funcionalidad/ **Consistencia** / Uniformidad de retorno al procesamiento / métrica
- 1.2.3.1 Funcionalidad/ **Consistencia** / Uniformidad del vocabulario y de la simbología utilizada / métrica
- 1.3.1.1 Funcionalidad/ **Corrección** / Correspondencia de las descripciones con los objetos / métrica
- 1.3.2.1 Funcionalidad/ **Corrección** / Funcionamiento correcto / métrica
- 1.3.3.1 Funcionalidad/ **Corrección** / Utilización correcta del idioma / métrica
- 1.4.1.1 Funcionalidad/ **Interoperabilidad** / Intercambiabilidad de datos / métrica
- 1.4.2.1 Funcionalidad/ **Interoperabilidad** / Intercambiabilidad en componentes e interfaces / métrica
- 1.5.1.1 Funcionalidad/ **Normalización** / Normalización de la simbología / métrica
- 1.5.2.1 Funcionalidad / **Normalización** / Normalización del vocabulario / métrica
- 2.1.1.1 Confiabilidad / **Madurez** / Frecuencia de fallas / métrica
- 2.2.1.1 Confiabilidad / **Recuperabilidad** / Opciones de recuperación / métrica
- 2.3.1.1 Confiabilidad / **Tolerancia de errores o fallos** / Procesamiento degradado / métrica
- 2.3.2.1 Confiabilidad/ **Tolerancia de errores o fallos** / Tratamiento de errores / métrica
- 3.1.1.1 Usabilidad / **Aprendibilidad** / Demo / métrica
- 3.1.2.1 Usabilidad / **Aprendibilidad** / Eficacia del Demo / métrica
- 3.1.3.1 Usabilidad / **Aprendibilidad** / Tutorial / métrica
- 3.1.4.1 Usabilidad / **Aprendibilidad** / Eficacia del Tutorial / métrica
- 3.1.5.1 Usabilidad / **Aprendibilidad** / Documentación / métrica
- 3.2.1.1 Usabilidad / **Atractividad** / Interacción atractiva / métrica
- 3.2.2.1 Usabilidad / **Atractividad** / Recuperación exitosa / métrica
- 3.2.3.1 Usabilidad / **Atractividad** / Tiempo de operación / métrica
- 3.3.1.1 Usabilidad / **Comprensibilidad** / Ayuda en línea / métrica
- 3.3.2.1 Usabilidad / **Comprensibilidad** / Interfaz de usuario adecuada / métrica
- 3.3.3.1 Usabilidad / **Comprensibilidad** / Terminología acorde al usuario / métrica
- 3.4.1.1 Usabilidad / **Difusión** / Amplitud / métrica
- 3.4.2.1 Usabilidad / **Difusión** / Frecuencia de operación / métrica
- 3.5.1.1 Usabilidad / **Operabilidad** / Operabilidad de las ayudas / métricas
- 3.5.2.1 Usabilidad / **Operabilidad** / Utilidad de las ayudas / métrica
- 4.1.1.1 Eficiencia / **Escalabilidad** / Disponibilidad / métrica.
- 4.2.1.1 Eficiencia / **Uso de los recursos** / Eficiencia en los recursos / métrica
- 4.3.1.1 Eficiencia / **Uso del tiempo** / Eficiencia en tiempo / métrica
- 5.1.1.1 Portabilidad / **Instalabilidad** / Módulo de instalación / métrica
- 5.1.2.1 Portabilidad / **Instalabilidad** / Documentación del módulo de instalación / métrica
- 5.1.3.1 Portabilidad / **Instalabilidad** / Módulo de configuración / métrica.
- 5.1.4.1 Portabilidad / **Instalabilidad** / Documentación del módulo de configuración / métrica.
- 5.2.1.1 Portabilidad / **Adaptabilidad** / Independencia del hardware / métrica.
- 5.2.2.1 Portabilidad / **Adaptabilidad** / Independencia del ambiente de software / métrica.
- 6.1.1.1 Calidad en uso / **Eficacia** / Eficacia de tareas / métrica.
- 6.1.2.1 Calidad en uso / **Eficacia** / Rendimiento de tareas / métrica.
- 6.2.1.1 Calidad en uso / **Productividad** / Eficiencia relativa al usuario / métrica.
- 6.2.2.1 Calidad en uso / **Productividad** / Proporción productiva / métrica.
- 6.3.1.1 Calidad en uso / **Satisfacción** / efectos psicológicos favoritos de usuario: SUMI métricas.

Fig. 2. Modelo MECHDAV completo.

Algunos ejemplos de cómo hacerlo serían: 1. Dividir la escala en dos categorías: insatisfactorio y satisfactorio. 2. Dividir la escala en cinco niveles o categorías obligatorias para un producto existente o alternativo: los niveles A, B, C, D, E (Figura 4). El nivel A es el del mejor caso, es aquel que sería el nivel Ideal por lograr, aunque con exceso de requerimientos. El nivel B es aquel que es considerado

alcanzable con los recursos disponibles. El nivel C es el situado para controlar que el nuevo sistema no se deteriora desde la situación presente. El nivel D es un límite para la aceptación del usuario, en el caso que el producto no cumpla con los niveles Normal y Recomendable; finalmente, el peor caso es el nivel E, donde el producto no cumple con requisitos mínimos de calidad (figura 4).

Característica / Subcaracterística / Atributo / métrica

- 1.3.1.1 **Funcionalidad/Corrección** / Correspondencia de las descripciones con los objetos métrica
- 1.3.3.1 **Funcionalidad/ Corrección** / Utilización correcta del idioma / métrica
- 1.5.1.1 **Funcionalidad/ Normalización** / Normalización de la simbología / métrica
- 1.5.2.1 **Funcionalidad / Normalización**/Normalización del vocabulario / métrica
- 3.1.1.1 **Usabilidad / Aprendizabilidad** / Demo / métrica
- 3.1.2.1 **Usabilidad / Aprendizabilidad** / Eficacia del Demo / métrica
- 3.1.3.1 **Usabilidad / Aprendizabilidad** / Tutorial / métrica
- 3.1.4.1 **Usabilidad / Aprendizabilidad** / Eficacia del Tutorial / métrica
- 3.1.5.1 **Usabilidad / Aprendizabilidad** / Documentación / métrica
- 3.2.1.1 **Usabilidad / Atractividad** / Interacción atractiva / métrica
- 3.3.1.1 **Usabilidad / Comprensibilidad** / Ayuda en línea / métrica
- 3.3.2.1 **Usabilidad / Comprensibilidad** / Interfaz de usuario adecuada / métrica
- 3.3.3.1 **Usabilidad / Comprensibilidad** / Terminología acorde al usuario / métrica
- 3.4.1.1 **Usabilidad / Difusión** / Amplitud / métrica
- 3.4.2.1 **Usabilidad / Difusión** / Frecuencia de operación / métrica
- 3.5.1.1 **Usabilidad / Operabilidad**/Operabilidad de las ayudas / métrica
- 3.5.2.1 **Usabilidad / Operabilidad** / Utilidad de las ayudas / métrica
- 5.1.1.1 **Portabilidad / Instalabilidad** / Módulo de instalación / métrica
- 5.1.2.1 **Portabilidad / Instalabilidad** / Documentación del módulo de instalación / métrica
- 5.1.3.1 **Portabilidad / Instalabilidad** / Módulo de configuración / métrica
- 5.1.4.1 **Portabilidad / Instalabilidad** / Documentación del módulo de configuración / métrica
- 6.3.1.1 **Calidad en uso / Satisfacción** / efectos psicológicos favoritos de usuario: SUMI métricas

Fig. 3. Subconjunto del modelo MECHDAV sugerido para la evaluación del producto por un usuario principiante.

Dado que en IEEE610 *Standard Glossary of Software Engineering Terms* se define la métrica como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”, se propone un conjunto de métricas básicas que dan la medida o pueden ser aplicadas para representar las características de calidad del tipo de software a evaluar.

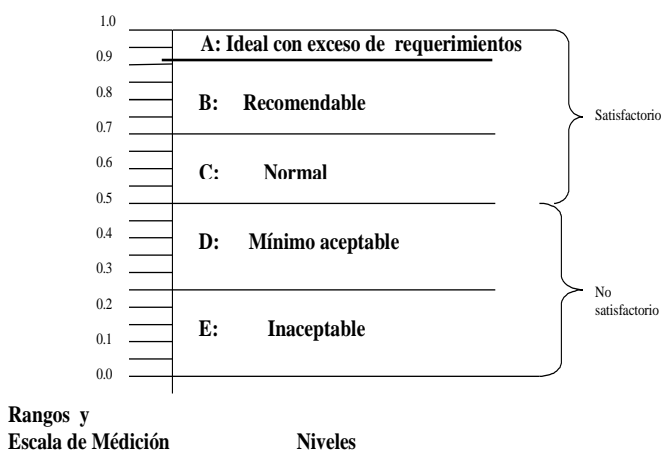


Fig. 4. Rangos y escala de medición para las métricas

La evaluación del software se realiza mediante una serie de cuestionamientos; al seleccionar una respuesta, ésta determinará las métricas de calidad en uso, las cuales son calificadas siguiendo el rango de la escala numérica para

calificar cada una de las métricas mostradas en la Tabla 1, ya sea mediante una ponderación (ecuación simple) o aplicada directamente.

Las descripciones y aplicaciones de las métricas están enfocadas hacia:

- La observación del comportamiento del software en cuanto a la diferencia entre los resultados de ejecución actuales y la especificación de los requerimientos (una vista de prueba y validación de la calidad).
- Ocurrencias inesperadas del comportamiento en el tiempo o utilización de recursos durante la operación del software.
- Las instrucciones directas al usuario para que realice una tarea específica, tomando nota de ciertos indicadores (por ejemplo: tiempo, número de ocurrencias de algún evento, etc.).
- Los cuestionamientos directos al usuario, en los cuales sólo se determina la existencia (presencia o ausencia) de la herramienta y/o la característica a analizar (atributos esenciales).

TABLA 1
INTERPRETACIÓN DE RANGOS DE NIVELES DE MÉTRICAS

VALOR	% CUMPLIMIENTO	SIGNIFICADO / INTERPRETACIÓN	RANGO
1.0	90 - 100	Excelente / Siempre	A
0.8	70 - 89	Satisfactorio / Casi siempre	B
0.6	50 - 69	Aceptable / Regularmente	C
0.4	30 - 49	Deficiente / En ocasiones	D
0.0	0 - 29	Inaceptable / Nunca o raras veces	E

Para respaldar este modelo se han desarrollado y documentado 44 métricas, con el formato que se muestra en de la Figura 5, más 11 que corresponden a la característica Calidad en Uso, subcaracterística Satisfacción, y atributo Efectos psicológicos de usuario (SUMI) [10]. La figura 6 muestra la sistematización de la métrica de la Figura 5 dentro de la herramienta.

Característica: 3. USABILIDAD.

Subcaracterística: 3.5 Operabilidad.

Atributo: 3.5.2 Operabilidad de las ayudas.

Métrica: 3.5.2.1 Relación de resultados exitosos mediante la facilidad de operación de las ayudas del software.

Objetivo: Determinar la relación del esfuerzo realizado por el usuario para obtener primero: resultados exitosos mediante la facilidad de operación de las ayudas del software.

Método: Analizar los componentes de cada herramienta para determinar la relación de resultados exitosos mediante la facilidad de operación de las ayudas del software.

Fórmula: $X = A/B$

Evaluación: $E(x) = \{ (0,0), (0.4, 40), (0.6, 60), (0.8, 80), (1,100) \}$

Medidas: A= Número de primeras tareas cuyos objetivos son completados exitosamente, utilizando las ayudas.

B= Número de tareas intentadas por el usuario.

Interpretación: Guía interactiva (intentos de usuario en tareas exitosas)

$0 \leq X \leq 1$; lo más cercano a 1 es lo mejor.

Fuente de referencia: nueva de MECHDAV, ISO 9126.

Fig. 5. Ejemplo de la documentación de una de las métricas utilizadas.

Plasmar los resultados tanto parciales como totales de la evaluación de la calidad del software no es tarea fácil, por lo que se deben elegir formatos simples y comprensibles para conseguir una valoración rápida y confiable de la calidad de las diferentes representaciones del software, por lo que se han elegido formatos tales como listas de comprobación (*checklist*), tablas simples de relación y matrices de control.

Las listas de comprobación (*checklist*) son cuestionarios donde se plasman preguntas (o aseveraciones) que deben ser contestadas (o confirmadas) plasmando uno de los valores dados en la escala acordada. Estos cuestionamientos en principio son hechos de manera tal que generen ideas (valoraciones). Se pueden utilizar para el control de todo el trabajo a realizar, y para cada fase en el que se encuentra la obra. Una matriz de control es una herramienta complementaria a todos los aspectos relacionados con el control de un proceso y que sirve para planificar y resumir el contenido y el desarrollo de un sistema de control. Normalmente incluye la variable de control (lo que se mide), la forma de medición, el lugar y el momento de la medición, el estándar, quién hace el análisis, quién actúa y cómo actuar. Las matrices de control son claves para el diseño del sistema de control, la implantación y el mantenimiento de los resultados obtenidos.

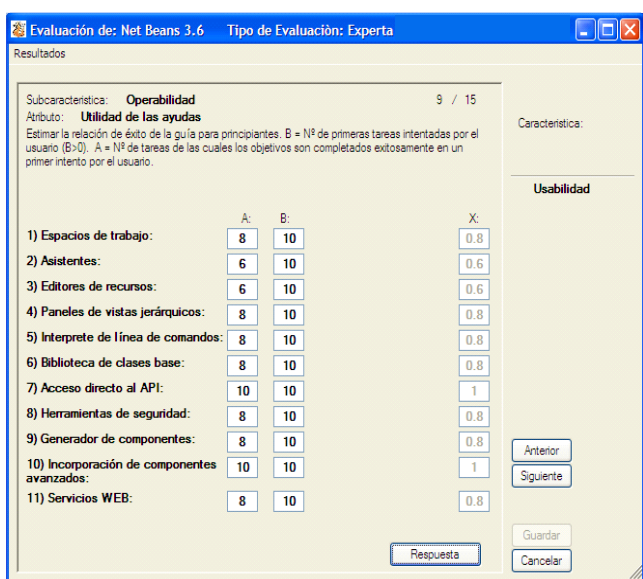


Fig. 6. Sistematización de la métrica de la Figura 5 dentro de la herramienta.

III. DISCUSIÓN DE LOS RESULTADOS

Este modelo se implementa en una herramienta computacional llamada con el mismo nombre que el modelo (MECHDAV) para sistematizar la evaluación técnica de cualquier ambiente visual que surja.

Los casos de estudio elegidos para las pruebas de las herramientas RAD son las plataformas visuales comerciales Visual Studio.Net, Net Beans y Eclipse, en diferentes versiones. Los resultados obtenidos por la evaluación técnica durante todo el proceso de medición realizado, en cada una

de las herramientas RAD de los paquetes de software comerciales mencionados, se muestran en las figuras 7, 8, 9 y 10.

Los modelos se implementan en la herramienta computacional propuesta, con la que es posible obtener un promedio de resultados (calificación) de diferentes evaluaciones de un mismo tipo (usuarios principiantes o usuarios expertos), realizadas a un producto de software por diversos evaluadores, para así tener una mayor precisión en la obtención del nivel de calidad alcanzado por el paquete de software.

En la Figura 7 se muestra un ejemplo de una fracción, de una de las matrices de control que se utilizan para obtener los resultados de la evaluación, de cada una de las herramientas de desarrollo de aplicaciones (RAD) en ambientes visuales, con respecto a cada una de las medidas que forman parte del modelo.

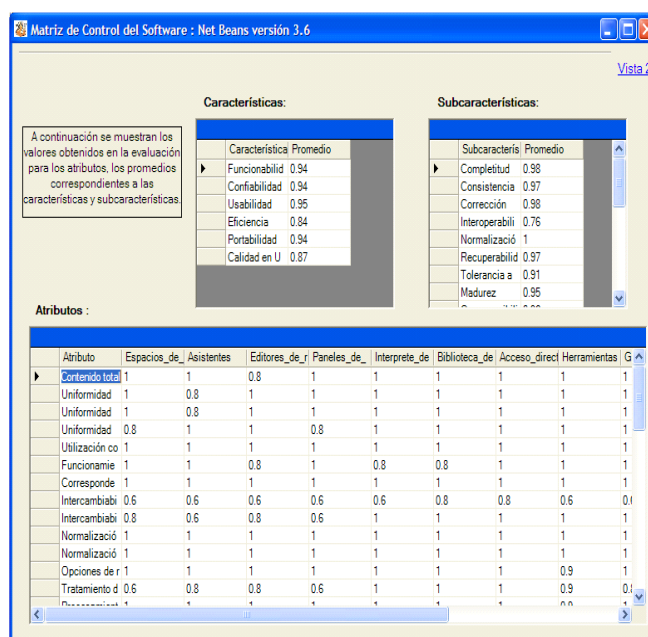


Fig. 7. Matriz de control que se genera con la evaluación del ambiente visual Net Beans, versión 3.6.

En este ejercicio se visualizan los resultados que arrojaron las evaluaciones del ambiente de desarrollo Net Beans versión 3.6, en cada una de las pruebas realizadas a este ambiente y a esta versión. Cuando se realizaron varias pruebas con diversos estudiantes, se detectó que los resultados no variaron considerablemente en la misma versión del mismo ambiente de desarrollo.

Cuando se obtienen los valores respectivos de la evaluación del software elegido, se genera el reporte final de la evaluación, donde se plasman los resultados definitivos y el porcentaje de cumplimiento. A su vez, se proporciona un esquema donde se muestran los puntos donde el software obtiene un buen nivel de clasificación de calidad, con qué criterio lo alcanza y además se brinda una recomendación final acerca de su aceptación o rechazo.

Con este sistema es posible obtener un promedio de resultados (calificación) de diferentes evaluaciones de un mismo tipo (usuarios principiantes o usuarios expertos), realizadas a un producto de software por diversos evaluadores, para así tener una mayor precisión en la obtención del nivel de calidad alcanzado por el paquete de software (Figura 10).

Subcaracterística / Atributo	Herramientas									
	ET	A	ER	PVJ	IA	LCB	ADA	HS	GG	GA
Complettud	1	1	1	1	1	1	0.8	0.8	1	1
Consistencia	1	1	1	1	1	1	1	1	1	0.8
Uniformidad de la estructura	1	1	1	1	1	1	1	1	1	1
Uniformidad de retorno al procesamiento	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
Corrección	1	1	1	1	1	1	1	1	1	1
Utilización correcta del idioma	1	1	0.9	1	0.9	1	1	0.8	0.9	0.9
Funcionamiento correcta de las funciones	1	1	1	1	1	1	1	1	1	1
Correspondencia de las descripciones con objetos	1	1	1	1	1	1	1	1	1	1
Interoperabilidad	1	1	1	1	1	1	1	1	1	1
Intercambiabilidad de datos	0.9	1	1	0.9	1	1	1	0.9	1	1
Intercambiabilidad en componentes de interfaces	1	1	1	1	0.8	1	1	1	1	1

Fig. 8. Matriz de control que se genera con la evaluación del ambiente visual Visual Studio.Net.

Características	Promedio	Nivel de Clasificación	Criterio	Conclusiones
Funcionabilidad	0,95	Excelente	(X) Sin modificación	(X) Aceptado
Confiabilidad	0,92	(X) Satisfactorio	Pequeñas modificaciones	Rechazado
Usabilidad	0,88	Aceptable	Grandes modificaciones	
Eficiencia	0,87	Deficiente		
Portabilidad	0,82	Inaceptable		
Calidad en Uso	0,9			
TOTAL:	0,89			

Tipo de Evaluación:
 Nombre del Ambiente:
 Descripción:

Nombre del Evaluador:
 Organización:
 Cargo:
 Área:

Fig. 9. Reporte técnico final de la evaluación del ambiente Visual Studio.NET, versión 2003

IV. CONCLUSIONES Y RECOMENDACIONES

Se realizaron seis pruebas para los ambientes visuales más populares y conocidos en el mercado, Visual Studio.Net Ver 7.0, Net Beans Ver. 3.6 y Eclipse 3.1.1, para tres usuarios

expertos (maestros del ITCM) y tres principiantes (alumnos de ingeniería en sistemas computacionales).

El Visual Studio.Net obtuvo una evaluación general promedio de 0.89 (89%) entre los usuarios principiantes y un puntaje de 0.88 (88%) entre los expertos, dentro del rango de valores propuesto. Una de sus mayores debilidades se ubica en su característica de portabilidad, lo cual es comprensible dada su dependencia de la plataforma Windows de Microsoft (Figura 5). El nivel de clasificación de calidad alcanzado es Satisfactorio sin recomendaciones, ya que no requiere de modificaciones en su diseño (sólo actualizaciones) y por lo tanto es aceptado ampliamente. El Net Beans y el Eclipse obtuvieron una evaluación general promedio de 0.92 (92%) y 0.93 (93%), respectivamente, entre los usuarios expertos que lo evaluaron.

Una de las mayores debilidades de Eclipse, y de Net Beans, versión 3.6, se ubica en su calidad en uso, dada la dificultad de su utilización por el usuario común por el aprendizaje obligado del lenguaje Java que requiere, ya que este paquete es dependiente de este lenguaje. El nivel de clasificación de calidad alcanzado es Excelente, sin recomendaciones, ya que no requiere de modificaciones en su diseño (sólo actualizaciones) por lo que se recomienda ampliamente.

Debido a los resultados obtenidos en su evaluación técnica, sin gran variabilidad en el porcentaje obtenido (sólo del 2%), y un nivel de clasificación de la calidad obtenida como Satisfactoria en Visual Studio.Net y Excelente en Net Beans y Eclipse, para unas versiones y Satisfactorio en otras (Figura 11), se requiere tomar una decisión acerca de la adquisición de alguno de estos ambientes, tomándose en cuenta otros parámetros importantes tales como los costos, la plataforma en la que se van a desarrollar otros sistemas para interactuar con estos ambientes, etc.

Ambiente Visual	Funcionabilidad	Confiabilidad	Usabilidad	Eficiencia	Portabilidad	Calidad en Uso	Promedio Total
Netbeans ver. 3.6 [Experta]	0.96	0.97	0.95	0.87	0.94	0.93	0.92
Visual Studio .NET ver. 2003 [Experta]	0.95	0.95	0.96	0.92	0.94	0.87	0.93
Net Beans versión 3.6 [Experta]	0.9	0.94	0.88	0.75	0.97	0.88	0.89
Net Beans versión 3.6 [Experta]	0.92	0.97	0.93	0.91	0.94	0.9	0.93
Net Beans versión 3.6 [Experta]	0.94	0.94	0.95	0.84	0.94	0.87	0.91
Promedio Totales	0.93	0.95	0.93	0.86	0.95	0.87	0.92

Nivel de Clasificación: (X) Excelente, Satisfactorio, Aceptable, Deficiente, Inaceptable
 Criterio: (X) Sin modificación, Pequeñas modificaciones, Grandes modificaciones
 Conclusiones: (X) Aceptado, Rechazado

Fig. 10. Reporte promedio para diversas evaluaciones de usuarios expertos para un mismo ambiente visual, Net Beans versión 3.6.

REFERENCIAS

Report of Software Evaluation: NetBeans 6.1 Evaluation Type: Expert. Date: Thursday, 05-28-20...

MECRAD

Type of Evaluation: Expert
 Environment Name: NetBeans 6.1
 Description: Ambiente de Desarrollo de Aplicaciones

Features	Average	Classification Level	Criterion	Conclusions
Functionality	0.90	<input type="checkbox"/> Excellent	<input checked="" type="checkbox"/> Without Modification	<input checked="" type="checkbox"/> Accepted
Reliability	0.90	<input checked="" type="checkbox"/> Satisfactory	<input type="checkbox"/> Little Modifications	<input type="checkbox"/> Rejected
Usability	0.80	<input type="checkbox"/> Acceptable	<input type="checkbox"/> Big Modifications	
Efficiency	0.80	<input type="checkbox"/> Deficient		
Portability	0.80	<input type="checkbox"/> Unacceptable		
Quality in Use	0.80			
TOTAL:	0.83			

Evaluator Name: Hector Rodrigo Ramirez Sanchez
 Organization: ITCM
 Position: Estudiante
 Area: Dpto. Sistemas y Computación

Exit

Fig. 11. Reporte técnico final de la evaluación realizada por un estudiante de postgrado del ambiente NetBeans, versión 6.1 en inglés.

Por lo anterior, esta propuesta sirve para evaluar y seleccionar las herramientas RAD de manera eficaz y fácil, sus plataformas de desarrollo (IDE) más adecuadas para crear aplicaciones en ambientes visuales, y para generar sistemas de información con calidad y sustentabilidad; por ello, por la calidad de sus herramientas RAD, cualquiera de los tres ambientes visuales analizados se considera muy recomendable técnicamente para generar sistemas de información. El sistema está dirigido a las instituciones de educación superior, organizaciones, empresas, usuarios finales, así como a los estudiantes de ingeniería, profesores de educación superior, entre otros, que se dediquen a construir aplicaciones para los sistemas de información en ambientes visuales.

Como trabajo futuro se recomienda hacer revisiones periódicas a MECHDAV- MECRAD para su actualización, así como tratar de introducir el modelo de evaluación técnica de las herramientas de desarrollo de aplicaciones en los ambientes visuales de los sitios WEB y en ambientes de otro tipo.

ACKNOWLEDGMENT

Los autores de este trabajo desean agradecer al Centro de Investigación en Computación (CIC) y al Instituto Politécnico Nacional (IPN), al ITESM-CCM, al ITCM-Tamaulipas, México, y a la Universidad Autónoma de Tamaulipas, México, por su apoyo para llevar a cabo el mismo.

- [1]. R. Pressman, Ingeniería de Software. Un enfoque práctico, McGraw Hill/Interamericana de España, S.A.U. Séptima Edición, 2005.
- [2]. A. Gutiérrez, Modelo de evaluación para el aseguramiento de la calidad del Software, Modelo MECA. Instituto Politécnico Nacional, 2003.
- [3]. A. Gutiérrez, "Metodología para el aseguramiento de la calidad del Software MACS". Instituto Politécnico Nacional, 1999.
- [4]. ISO9241, Ergonomics of human-system interaction; Part 151 Guidance on World Wide Web user interfaces; and Industry Standards for User Centered Design, October 2000, www.usability.serco.com/trump, 2008.
- [5]. IEEE610, Software Engineering Standards Collection, Standard Glossary of Software Engineering Terminology. IEEE, Std 610.12-190, 1994.
- [6]. IEEE1061:1992, Software Quality Metrics Methodology, IEEE 1061 Standard for a Software Quality Metrics Methodology, 1992.
- [7]. ISO/IEC9126, Software Product Evaluation; Parte 1: Quality, Characteristics and Guidelines for their Use; Parte 2: Métricas externas para una validación de la calidad de software; Parte 3: Métricas externas para una validación de la calidad de software, 1997.
- [8]. ISO/IEC14598, Information Technology, Software Product Evaluation. (Partes 1, 2, 3, 4, 5), 1998.
- [9]. ISO/IEC 25000 "SQuARE System and Software Quality Requirements and Evaluation", JTC C1/SC7/ WG6 N2246. Plan y configuración de los requerimientos de calidad de software y evaluación. SQUARE2000, 2005.
- [10]. SUMI 2000, Human Factors Research Group, 2000. SUMI: Software Usability Measurement Inventory, European Directive on Minimum Health and Safety Requirements for Work with Display Screen Equipment (90/270/EEC). Ireland, 2000.
- [11]. M. Moreno, "Aplicación de las Métricas de Calidad del Software en la Evaluación Objetiva de Gramáticas Independientes de Contexto Inferidas". I Simposio Avances en Gestión de Proyectos y Calidad del Software. Salamanca, España, pp. 209-220, 2004.
- [12]. L. Olsina, "Medición y Evaluación de Calidad en Uso: Un Caso de Estudio para una Aplicación E-Learning". 9º Taller Iberoamericano de Ingeniería de Requisitos y Ambientes de Software, La Plata, Argentina, pp. 317-330, ISBN-10:950-34-0360, 2006.
- [13]. M. Piattini, "Métricas para la Evaluación de Modelos de Proceso de Negocio". 9º Taller Iberoamericano de Ingeniería de Requisitos y Ambientes de Software, La Plata, Argentina, pp. 419-432, ISBN-10:950-34-0360; pp. 419-43, ISBN-10:950-34-0360, 2006.
- [14]. O. Pastor, "Evaluación de la Usabilidad en un Entorno de Arquitectura Orientadas a Modelos". 9º Taller Iberoamericano de Ingeniería de Requisitos y Ambientes de Software, La Plata, Argentina, pp. 331-344, ISBN-10:950-34-0360, 2006.
- [15]. L.S. Vargas-Pérez, A.F. Gutiérrez-Tornès, E.M. Felipe-Riverón. "MECRAD: Model and Tool for the Technical Quality Evaluation of Software Products in Visual Environment". ICCGI-5.2, 4th International Conference on Wireless and Mobile Communications (ICWMC 2008) and 3rd International Multi-Conference on Computing in the Global Information Technology (ICCGI 2008). Product Number E3275. BMS Part Number CFP0840B-CDR. ISBN 978-0-7695-3275-2. Library of Congress Number 2008926137 pp. 107-112. IEEE Computer Society. IARIA. Athens, Greece, 2008.