

Tipo de artículo: Artículo original  
Temática: Soluciones Informáticas  
Recibido: 10/12/17 | Aceptado: 10/01/18 | Publicado: 26/01/18

## **Métricas de usabilidad para la detección de errores en el diseño y funcionamiento del entorno de escritorio de la Distribución de GNU/LINUX NOVA**

### ***Usability metrics for the detection of errors in the design and operation of the desktop environment of the GNU / LINUX NOVA distribution***

Yasmary Prieto Carmona<sup>1</sup>, Juan Fuentes Bauta<sup>2</sup>

<sup>1</sup> Universidad de las Ciencias Informáticas. [yasmary@uci.cu](mailto:yasmary@uci.cu),

<sup>2</sup> Universidad de las Ciencias Informáticas. [Informaticas,jbauta@uci.cu](mailto:Informaticas,jbauta@uci.cu)

\* Autor para correspondencia: [yasmary@uci.cu](mailto:yasmary@uci.cu),

---

#### **Resumen**

La aceptación de los sistemas informáticos es determinada por las personas que finalmente lo utilizarán, si los mismos no son fáciles de usar, los usuarios no los quieren y en el peor de los casos no serán utilizados. Por lo que no es bueno ocuparse de la facilidad de uso de un producto una vez liberado. El objetivo de este trabajo es definir qué criterios pueden ser utilizados por los programadores y evaluadores de software para medir la facilidad de uso del entorno de escritorio de la Distribución Cubana de GNU/LINUX NOVA de forma que contribuyan a la detección temprana de errores de funcionamiento y de diseño. En esta investigación se analizan los factores de fracaso en el desarrollo de sistemas de información y las ventajas que poseen los proyectos de software libre al tener al usuario final como parte del equipo de desarrollo. Finalmente se hace una propuesta de métricas para evaluar la usabilidad basadas en el modelo de calidad de software libre del Centro de Nacional de Desarrollo e Investigación en tecnologías libres. La usabilidad es una condición necesaria para la supervivencia de productos de software, si estos no son fáciles de usar la gente no los quiere

**Palabras clave:** distribución linux; usabilidad; factor de éxito

#### ***Abstract***

*The acceptance of computer systems is determined by the people who will finally use it, if they are not easy to use, users do not want them and in the worst case they will not be used. So it is not good to deal with the ease of use of a product once released. The objective of this paper is to define what criteria can be used by programmers and software evaluators to measure the ease of use of the desktop environment of the GNU/LINUX NOVA Cuban Distribution in order to contribute to the early detection of operating errors and of design. This research analyzes the factors of failure in the development of information systems and the advantages of free software projects by having the end user as part of the development team. Finally, a proposal of metrics is made to evaluate the usability based on the free software quality model of the National Center for Development and Research in free technologies. The usability is a necessary condition for the survival of software products, if they are not easy to use, people do not want them*

**Keywords:** *linux distribution; usability; success factor*

---

## **Introducción**

En la Universidad de las Ciencias Informáticas (UCI) se desarrolla la Distribución de GNU/LINUX Nova para satisfacer las necesidades de la Migración a Plataformas de Código Abierto como parte del proceso de Informatización de la Sociedad. Es la Distribución recomendada para ser utilizada como tecnología base de despliegue y desarrollo de aplicaciones informáticas en los Organismos de la Administración Central del Estado. Ofrece a los usuarios la posibilidad de personalización, a la medida de sus necesidades, teniendo en cuenta las características tecnológicas de nuestro país para computadoras de alta y bajas prestaciones. Por el acuerdo No.084 del 2004 del Comité Ejecutivo del Consejo de Ministros se identificó la necesidad de ejecutar acciones que garantizaran la migración ordenada y progresiva hacia aplicaciones y plataformas de código abierto, en concordancia con el desarrollo del proceso de informatización de la sociedad como parte de la ejecución por el país de una política orientada a alcanzar la seguridad, invulnerabilidad e independencia tecnológica. Los factores que dificultan el proceso migratorio actual son: “la poca comprensión de los usuarios de la importancia del proceso de migración, los errores humanos cometidos por los especialistas y la poca documentación de las condiciones iniciales de la entidad que afecta la planificación de una acertada estrategia de migración a la medida” [1]. “La carencia de programas informáticos, soportados en plataforma Linux y capaces de cubrir las necesidades de automatización de los sistemas contables, torna polémico e inviable el proceso migratorio en este ámbito. Los usuarios, por su parte, ofrecen no poca resistencia al cambio, ya que se encuentran mucho más familiarizados con la plataforma Windows e inmersos en un ambiente de intercambio cooperativo de software propietario, el cual no está sujeto a sanción alguna por concepto de violación de leyes y derechos de autor. La mayoría de los informáticos del país están formados y aún trabajan sobre plataforma

Windows; esto ocurre a pesar de que se ha incrementado la presencia de Linux (en alguna de sus distribuciones) a nivel de servidores, pero esta plataforma no abunda en las estaciones de trabajo”[2]. Además de estos inconvenientes, durante la II Conferencia Científica de la UCI 2017 se abordaron en el taller de software libre algunas dificultades que actualmente enfrentaba el proceso de migración. Poco personal para la capacitación de algunos servicios telemáticos, problemas organizativos por parte del cliente, resistencia al cambio no solo por la costumbre del uso de tecnologías privativas sino por las experiencias con versiones anteriores del producto, inconformidades de los usuarios con algunos aspectos de la interfaz gráfica del producto. Teniendo en cuenta estas dificultades el objetivo de este trabajo es recopilar criterios que permitan corregir errores tempranos tanto de diseño y funcionamiento de manera que contribuya no solo a la aceptación del software por parte de los usuarios finales, sino que favorezca al mismo proceso de migración que se está llevando paulatinamente en varias entidades del país.

## **Materiales y métodos o Metodología computacional**

Usabilidad según diferentes autores.

Este término usabilidad es dado por Jakob Nielsen para referirse a la facilidad de uso de las interfaces de usuario. Puede ser comprendida con más claridad a partir de las siguientes variables [3]:

- **Facilidad de Aprendizaje:** ¿Cómo de fácil resulta para los usuarios llevar a cabo tareas básicas la primera vez que se enfrentan al diseño?
- **Eficiencia:** Una vez que los usuarios han aprendido el funcionamiento básico del diseño, ¿cuánto tardan en la realización de tareas?
- **Cualidad de ser recordado:** Cuando los usuarios vuelven a usar el diseño después de un periodo sin hacerlo, ¿cuánto tardan en volver a adquirir el conocimiento necesario para usarlo eficientemente?
- **Eficacia:** Durante la realización de una tarea, ¿cuántos errores comete el usuario? ¿cómo de graves son las consecuencias de esos errores?, ¿cómo de rápido puede el usuario deshacer las consecuencias de sus propios errores?
- **Satisfacción:** ¿Cómo de agradable y sencillo le ha parecido al usuario la realización de las tareas?

Ian Sommerville: un requerimiento del producto no funcional 3 que se puede verificar [4].

Roger Pressman: es una característica relevante que proporciona una base útil para evaluar la calidad de los sistemas basados en la web [5].

La norma ISO 9241 define usabilidad como: “el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto

de uso” [6].

La norma ISO 9126 define usabilidad como: “la capacidad del producto software de ser aprendido, usado y atractivo para el usuario, cuando se utiliza en condiciones específicas” [7].

ISO/IEC 25000 define usabilidad como: “capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones” [8]

Estos conceptos tienen en común al usuario, por lo que la calidad de su experiencia al interactuar con un producto o sistema, ya sea un sitio web, una aplicación de software, tecnología móvil o cualquier dispositivo es relevante.

¿Por qué es importante la usabilidad?

Según Nielsen en la web, “la usabilidad es una condición necesaria para la supervivencia. Si un sitio web es difícil de usar, la gente se va. Si la página no expone claramente lo que una empresa ofrece y lo que pueden hacer los usuarios en el sitio, la gente se va. Si los usuarios se pierden en un sitio web, se van. Si la información de un sitio web es difícil de leer o no responder a las preguntas claves de los usuarios, se van” [3]. Por lo que la usabilidad juega un papel importante en el desarrollo de sistemas interactivos y los sistemas operativos son sistemas interactivos.

### **Factores de éxito o fracasos del sistema de información**

Los siguientes factores [9] son algunos de muchos más que influyen en el éxito o fracaso de un sistema de información:

- La ausencia de capacitación adecuada
- El rol de los usuarios en el proceso de implementación.
- La calidad de administración del proceso de implementación.

Al hablar del segundo factor de éxito o fracaso se puede decir que:

“La participación de los usuarios en proyectos de software ha sido considerado un requisito previo para el éxito de los sistemas informáticos” [9], sin embargo, con frecuencia a los usuarios se “les deja fuera del proceso de implementación, como no pueden entender lo que están haciendo los técnicos concluyen que es mejor dejar todo el proyecto en manos de los profesionales de sistemas”.

Resultados de la participación del usuario en el desarrollo y la ventaja de los ambientes colaborativos

Los resultados que traerá la participación del usuario en el proceso de implementación serán positivos, sobre todo si se encuentran involucrados desde el diseño y así “en primer lugar tendrían más oportunidades de moldear el sistema según sus prioridades y requerimientos de negocios y más oportunidades de controlar el producto resultante.

En segundo lugar, es probable que reaccionen de manera positiva al sistema terminado porque han sido participantes activos en el proceso” [9]

Según Fitzgerald, el 24% de los proyectos de software libre y/o código abierto se inician por necesidad de una compañía, Esto indica que el 76% de los proyectos se inician por necesidad de los mismos desarrolladores, por lo que se puede llegar a la conclusión que la mayoría de las veces, los desarrolladores son los mismos usuarios finales [10].

Un estudio realizado demuestra que las sugerencias de los usuarios generaron más del 20% de los cambios en casi el 50% de los proyectos de código abierto seleccionados para el estudio, demostrando esto el importante aporte que realizan los usuarios al proceso de desarrollo en software libre y/o código abierto [11].

El proceso de desarrollo de software libre se caracteriza por tener un desarrollo en red. Es decir, está orientado a unas prácticas que posibiliten la creación de software a personas que están dispersas, pero que pueden comunicarse mediante Internet [12].

Esto posibilita que puedan acceder al código fuente, permitiendo disminuir el tiempo de desarrollo, escribir código, probar y corregir errores de forma paralela [11].

A continuación se muestran algunos de los criterios que aportarán información sobre la usabilidad del sistema operativo, en este caso basados en el modelo para el Aseguramiento de Calidad en el Desarrollo de Software Libre de la fundación CENDITEL<sup>1</sup>

Este modelo tiene como objetivo la mejora continua del proceso de desarrollo y plantea realizar una evaluación a las prácticas llevadas a cabo durante el desarrollo de una aplicación de software a fin de verificar si dichas prácticas se corresponden con buenas prácticas de desarrollo características del área de la ingeniería del software y del ámbito de desarrollo de software libre. Los criterios que se presentan a continuación son derivados de este modelo en su primera versión [13], basándose la misma en cuanto a los indicadores de usabilidad en la ISO/IEC 9126.

## Resultados y discusión

Tabla1. Criterios para evaluar la capacidad del sistema de ser entendido

Métrica	Método de medición
Funciones evidentes	Contar las funciones evidentes al usuario y comparar con el número total de funciones
Función de entendibilidad	Contar el número de funciones cuyos propósitos son entendidos por los usuarios y comparar con

<sup>1</sup> Centro Nacional de Desarrollo e Investigación en tecnologías libres

el número de funciones de la interfaz

Método de cálculo para el criterio de funciones evidentes:

$$x=A/B$$

donde

A: Número de funciones (o tipos de funciones) evidentes al usuario

B: Total de funciones (o tipos de funciones)

Interpretación del valor medido

$$0 \leq X \leq 1$$

Valores cercanos a 1 indican mayor cantidad de funciones del software que son evidentes al usuario

Método de cálculo para el criterio de función de entendibilidad:

$$x=A/B$$

Donde

A: Número de funciones de la interfaz cuyos propósitos son entendidos por el usuario

B: Total de funciones de la interfaz

Interpretación del valor medido

$$0 \leq X \leq 1$$

Valores cercanos a 1 indican mayor cantidad de funciones del software cuyos propósitos son entendidos por el usuario

Tabla 2. Criterios para evaluar la capacidad del sistema de ser aprendido

Métrica	Método de medición
Existencia completa de la documentación de usuario y/o ayudas	Contar el número de funciones implementadas con ayudas y/o documentación y comparar con el número total de funciones en el software

Método de cálculo para este criterio:

$$x=A/B$$

Donde

A: Número de funciones que se encuentran descritas.

B: Número total de funciones

Interpretación del valor medido

$$0 \leq X \leq 1$$

Valores cercanos a 1 indican mayor cantidad de funciones del software descritas en la documentación

Tabla 3. Criterios para evaluar la efectividad

Métrica	Método de medición
Tareas completadas	Contar la proporción de tareas que llegan a ser realizadas por el usuario

Método de cálculo para el criterio de tareas completadas:

$$X = \sum Ai/B$$

Donde

$A_i$  : Número de tareas completadas por el Usuario  $i$

B: Número de tareas establecidas, multiplicado por el número total de usuario

Interpretación del valor medido

$$0 \leq X \leq 1$$

Valores cercanos a 1 indican mayor cantidad de tareas completadas por los usuarios definidos para la prueba

Tabla 4. Criterios para evaluar la satisfacción

Métrica	Método de medición
Actitud positiva ante el	Realizar la siguiente pregunta a cada uno de los usuarios participantes en la prueba: ¿cómo

software	evaluaría su actitud positiva frente al software? La pregunta será evaluada de acuerdo a la siguiente escala de intervalos entre el cero (0) y uno (1): a) (0,8 - 1] ->Muy Alto b) (0,6 - 0,8] ->Alto c) (0,4 - 0,6] ->Medio d) (0,2 - 0,4] ->Bajo e) [0 - 0,2] ->Muy Bajo
Satisfacción de los usuarios del software en comparación con el software anterior	Realizar la siguiente pregunta a cada uno de los usuarios participantes en la prueba: ¿cuál es su grado de satisfacción con el software al compararlo con el software anterior que utilizaba para las mismas tareas?
Aporte a la productividad por el uso del software	Realizar la siguiente pregunta a cada uno de los usuarios participantes en la prueba: ¿aumenta su productividad respecto a las tareas que realiza con el software?

Método de cálculo para el criterio **Actitud positiva ante el software:**

$$x=A / B$$

Donde

A: Sumatoria de los valores cuantitativos dados por los usuarios para la pregunta realizada

B: Número total de usuarios

Interpretación del valor medido

$$0 \leq X \leq 1$$

Valores cercanos a 1 indican más actitudes positivas de parte de los usuarios ante el software

Método de cálculo para el criterio **Satisfacción de los usuarios del software en comparación con el software anterior:**

$$x=A / B$$



Donde

A: Sumatoria de los valores cuantitativos dados por los usuarios para la pregunta realizada

B: Número total de usuarios

Interpretación del valor medido

$$0 \leq X \leq 1$$

Valores cercanos a 1 indican mayor satisfacción de los usuarios con respecto al software

Método de cálculo para el criterio **Aporte a la productividad por el uso del software:**

$$x=A / B$$

Donde

A: Sumatoria de los valores cuantitativos dados por los usuarios para la pregunta realizada

B: Número total de usuarios

Interpretación del valor medido

$$0 \leq X \leq 1$$

Valores cercanos a 1 indican mayor productividad de los usuarios al utilizar el software

Tabla 5. Criterios para evaluar la eficiencia

Métrica	Método de medición
Tiempo para realizar tareas en comparación con un experto	medir el tiempo que cada usuario tarda en realizar todas las tareas y promediarlo, luego comparar con el tiempo que le toma a un experto realizar la misma prueba

Método de cálculo para este criterio:

$$x=A - B$$

Donde

A: Tiempo que le toma a un experto completar la prueba

B: Tiempo promedio que le toma a los usuarios completar la prueba: sumatoria de los tiempos de cada usuario dividido entre el número total de usuarios

Interpretación del valor medido

$$-\infty \leq X \leq +\infty$$

Valores positivos de X indican que el software es más fácil de utilizar. El rango de valores negativos aceptables será definido por el experto

A partir del valor medido de cada una de las métricas se podrá saber en qué medida se cumplen cada una de ellas en el entorno de escritorio contribuyendo así a conocer en los casos negativos el por qué se obtiene ese valor.

## Conclusiones

1. Con frecuencia se deja a los usuarios fuera del proceso de implementación, sin embargo, este es uno de los factores que influye en el éxito de un producto de software.
2. Una de las fortalezas que tiene el ambiente colaborativo es que los mismos desarrolladores son usuarios finales lo que contribuye favorablemente a la realización de pruebas durante el desarrollo.
3. La usabilidad es una condición necesaria para la supervivencia de productos de software, si estos no son fáciles de usar la gente no los quiere
4. El modelo para el aseguramiento de la calidad de la fundación CENDITEL facilita una serie de criterios que pueden ser aplicados en el contexto del software libre en Cuba ya que está diseñado específicamente para este tipo de ambientes

## Referencias

- [1] Villazon Yoandy. (2013). El proceso de migración a aplicaciones de código abierto en Cuba desde un enfoque metodológico. *Revista Cubana de Ciencias Informáticas*; Vol. 7, Núm. 4 (2013): Octubre–Diciembre. Recuperado de <http://rcci.uci.cu/?journal=rcci&page=article&op=view&path%5B%5D=287>

- [2] Montes de Oca Montano, J. L. (2015). La migración hacia software libre en Cuba: complejo conjunto de factores sociales y tecnológicos en el camino de la soberanía nacional. *Revista Universidad y Sociedad* [seriada en línea], 7 (3). pp. 119-125. Recuperado de <http://rus.ucf.edu.cu/>
- [3] Nielsen, J. (2003). Usability 101: Introduction to Usability, UseIt.com Alertbox. Disponible en: <http://www.useit.com/alertbox/20030825.html>
- [4] Ian Sommerville. *Ingeniería de Software*. PEARSON EDUCACION, Madrid 2005. 172
- [5] Roger S Pressman. *Ingeniería de Software: Un Enfoque Práctico*. Quinta Edición, 2002. 589
- [6] ISO 9241-11. (1998). Ergonomic requirements for office work with visual display terminals (VDT)s-Part 11 Guidance on usability, 1998
- [7] ISO/IEC 9126: “Software Engineering - Product quality”, International Organization for Standardization, 2000
- [8] ISO/IEC 25010. Disponible en: <http://iso25000.com/index.php/normas-iso-25000/iso-25010>
- [9] Panizzi, M. 2015. Propuesta de Recomendaciones para la Implementación de Sistemas Informáticos Basadas en el Enfoque Socio-técnico y el Diseño Participativo. *Revista Latinoamericana de Ingeniería de Software*, 3(1): 1-40, ISSN 2314-2642
- [10] BRIAN FITZGERALD, 2006, The Transformation of Open Source Software. [online]. 2006. Vol.30, no.3. Available from: <http://www.jstor.org/stable/25148740>
- [11] SEBASTIAN ELBAUM and LUYIN ZHAO, 2003, Quality assurance under the open source development model. 2003. No.66
- [12] GREGORIO ROBLES MARTÍNEZ, JUAN JOSÉ IGLESIAS and ISRAEL HERRAIZ TABERNERO, 2007, Desarrollo de proyectos de software libre. February 2007
- [13] Modelo para el Aseguramiento de Calidad en el Desarrollo de Software Libre. 2012, Centro de Nacional de Desarrollo e Investigación en tecnologías libres