

**Título: Método para la medición de la calidad de software a los productos del proyecto @tis.**

**Autores:**

**MSc. Yordanka Barrios González.**

**Ing. Fidel Pérez Candelario.**

**Ing. Lenz Mesa Castillo**

**Institución: Universidad de Matanzas "Camilo Cienfuegos" dirección postal:**

**teléfono: 261013**

**dirección electrónica:**

[yordanka.barrios@umcc.cu](mailto:yordanka.barrios@umcc.cu)

[fidel.perez@umcc.cu](mailto:fidel.perez@umcc.cu)

[lenz.mesa@umcc.cu](mailto:lenz.mesa@umcc.cu)

## Resumen:

Uno de los problemas que se afrontan actualmente en la esfera de la computación es la calidad del *software*. Este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de *softwares*, los cuales han realizado gran cantidad de investigaciones al respecto con dos objetivos fundamentales:

1. ¿Cómo obtener un software con calidad?
2. ¿Cómo evaluar la calidad del software?

Esta investigación tiene como objetivo dar respuesta a la segunda pregunta a través de un método para la medición de la calidad de software a los productos del proyecto @tis, los cuales son desarrollados siguiendo la metodología Rational Unified Process(RUP) que debido a sus características se acopla perfectamente a evaluaciones de calidad en cada iteración de la etapa de prueba ( lo que pone en las manos de analistas y desarrolladores las deficiencias en tiempo para ser corregidas ) y evaluaciones en el producto final.

La calidad en la esfera informática es un problema con una complejidad superior a la de cualquier otra esfera. En ningún otro campo pueden tener mayor impacto los programas de calidad que en la industria del software. Este es un sector donde el concepto de calidad ha generado la revolución más radical. Es una actividad relativamente joven con relación a otras áreas productivas y presenta como peculiaridad un crecimiento en demanda y complejidad a mayor velocidad que las metodologías, el personal capacitado y las herramientas para automatizar la producción.

Uno de los problemas que se afrontan actualmente en la esfera de la computación es la calidad del *software*. Desde la década del 70, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de *softwares*, los cuales han realizado gran cantidad de investigaciones al respecto con dos objetivos fundamentales:

3. ¿Cómo obtener un *software* con calidad?
4. ¿Cómo evaluar la calidad del *software*?

Ambas interrogantes conllevan amplias respuestas, pero están estrechamente ligadas con el concepto de la calidad del *software*, que es el resultado de la primera y la fuente de la segunda.

Estas circunstancias han producido una prolongada crisis del software, donde los productos se entregan con demoras y en muchas ocasiones no responden a los requerimientos funcionales expresados por los usuarios.

Debido a esto nos hemos dado a las tareas de estudiar las métricas de calidad de software existentes para con esto llegar a nuestra propia definición de calidad de software, y con la misma medir nuestros productos de software en la Unidad Docente de Cárdenas "José A. Echevarría", donde opera el proyecto @tis.

Pretendemos adecuar los diseños de procedimientos para la medición de la calidad de software a nuestros productos y después de la realización de encuestas a los usuarios de las empresas implicadas medir nuestra eficiencia.

La gran parte de la literatura consultada por el autor se refiere a los términos de calidad de software como el grado con que el sistema cumple los requerimientos especificados y las necesidades o expectativas de los clientes o usuarios.

Otra de las definiciones lo caracterizan por ciertos atributos: fiabilidad, flexibilidad, robustez, comprensión, adaptabilidad, modularidad, complejidad, portabilidad, usabilidad, reutilización y eficiencia que determinan su utilidad y existencia.

### ¿QUE ES LA CALIDAD DEL SOFTWARE?

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

Modelo de medición de la calidad de software:

Modelo de calidad ISO 9126 el cual se define como el conjunto de características y sub-características y de como se relacionan entre sí.

Funcionalidad: Capacidad del producto de software para proveer funciones que cumplan con necesidades específicas o implícitas, cuando el software es usado bajo ciertas condiciones. Dentro de esta característica se manifiestan las subcaracterísticas: Adecuación , Corrección, Interoperabilidad, Seguridad, Conformidad.

Fiabilidad: Capacidad del producto software para mantener un nivel alto especificado de rendimiento cuando es utilizado bajo condiciones específicas. Subcaracterísticas: Madurez, Tolerancia a Fallos, Recuperabilidad.

Usabilidad: Capacidad del producto de software para ser atractivo, entendido, aprendido y utilizado por el usuario bajo condiciones específicas. Subcaracterísticas: Aprendibilidad, Comprensibilidad, Operabilidad, Atractividad.

Eficiencia: Capacidad del producto de software para proveer un rendimiento apropiado, relativo a la cantidad de recursos utilizados, bajo condiciones específicas. Subcaracterísticas: Comportamiento Temporal, Utilización de los recursos.

Mantenibilidad: Capacidad del producto para ser modificado. Subcaracterísticas: Analizabilidad, Cambiabilidad, Estabilidad, Facilidad de pruebas.

Portabilidad: Capacidad del producto software para ser transferido de un ambiente a otro. Subcaracterísticas: Adaptabilidad, Instalabilidad, Coexistencia, Reemplazabilidad.

### ¿COMO OBTENER UN SOFTWARE DE CALIDAD?

La obtención de un *software* con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del *software* que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del *software*.

Está claro que para que un producto software tenga una calidad óptima debe cumplir todas las características y subcaracterísticas antes expresadas. Existen modelos que se enfocan a medir la calidad del proceso de producción de software y otros que se orientan al producto final, el cual ha sido el escogido en nuestro caso, partiendo de la premisa si un software tiene "buena calidad" quiere decir que su proceso de desarrollo tuvo un buen curso. La metodología elegida para el análisis y diseño de nuestro sistema la constituye el Rational Unified Process (RUP) la cual parte desde la captura de los requerimientos, pasando por el análisis, diseño, implementación y prueba para lograr un sistema de acuerdo a los requerimientos iniciales de los usuarios.

Características de RUP:

- Orientado a la arquitectura
- Guiado por casos de usos
- Iterativo e incremental

Por la solidez de esta metodología hemos decidido evaluar nuestro producto de software en las etapas de prueba e implantación.

## ¿COMO CONTROLAR LA CALIDAD DEL SOFTWARE?

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como bien plantea Tom De Marco, "usted no puede controlar lo que no se puede medir".

Para la medición de la calidad del producto final proponemos la realización de encuestas a los usuarios los cuales son los mayores afectados si el software no cumple los requerimientos iniciales.

Las encuestas estarán integradas por afirmaciones en las cuales los usuarios deben dar una evaluación al software y señalar el peso que tiene esa afirmación para ellos en el producto de software.

Proponemos una encuesta donde estén presentes cada una de las sub-características incluida las característica que debe poseer el software para con esto evaluar su comportamiento.

La jerarquía utilizada se muestra en la fig.1 donde se representan las dimensiones con sus diferentes atributos, estos últimos tienen correspondencia directa con los items de la encuesta, por ese motivo no se especifican en este momento:

- D1: Funcionalidad
- D2: Confiabilidad
- D3: Usabilidad
- D4:Eficiencia
- D5:Mantenibilidad
- D6:Portabilidad

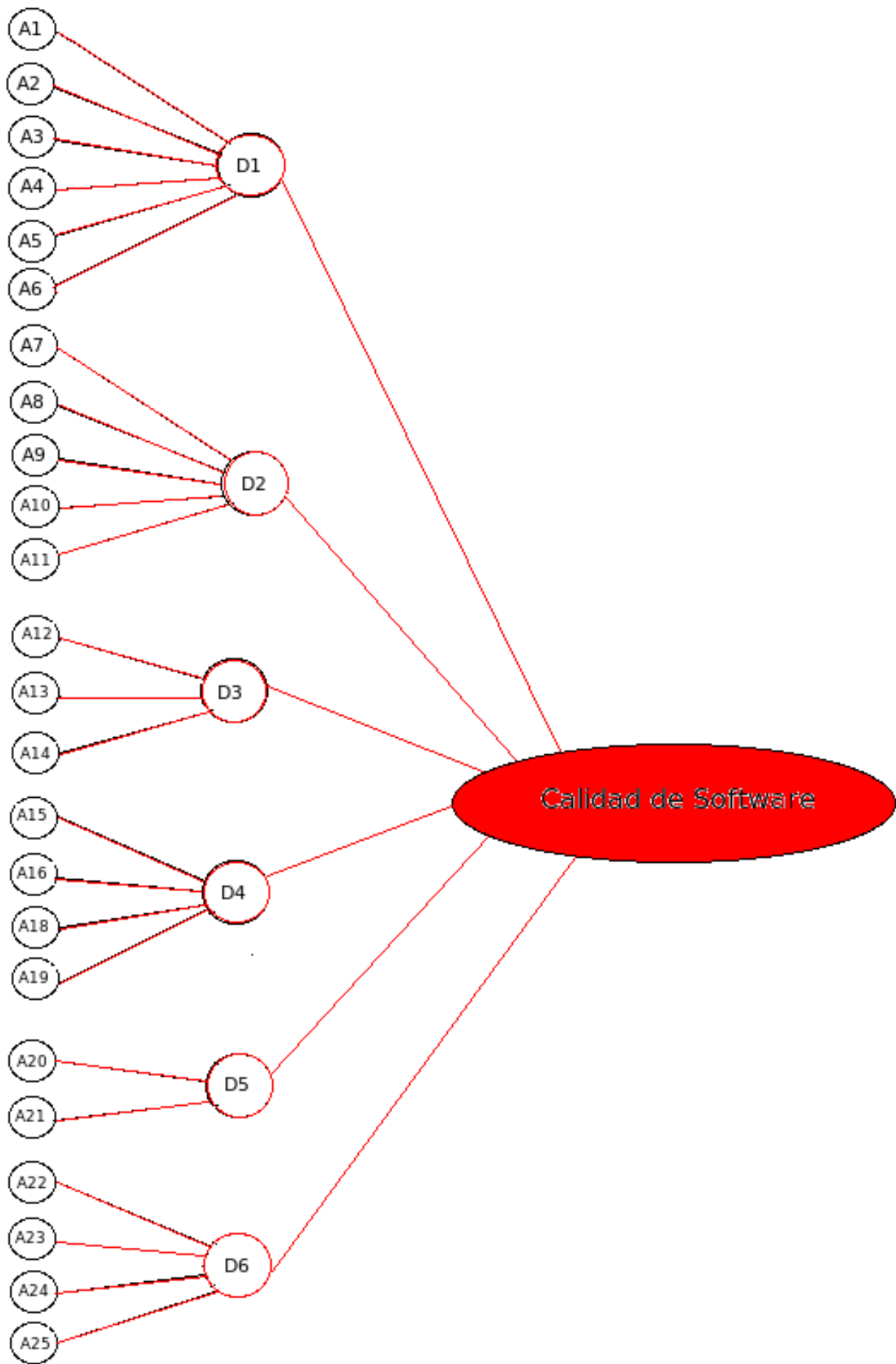


fig.1 Dimensiones con sus atributos.

Estimado Sr. o Sra.: A continuación se realizan una serie de afirmaciones que caracterizan la calidad del software. Usted deberá expresar su grado de acuerdo o desacuerdo con ellos deben estar presentes a la hora de medir la calidad de un software. Utilice para ello la siguiente escala:

- 5----->Totalmente de Acuerdo----->TA
- 4----->De acuerdo----->A
- 3----->Ni de Acuerdo ni en Desacuerdo----->N
- 2----->En desacuerdo----->D
- 1----->Totalmente en desacuerdo----->TD

Marque con una X el valor de la escala escogido por usted.

Complete, por favor la siguiente información:

Nombre del Software que utiliza: \_\_\_\_\_

Proveedor: @tis

Empresa: \_\_\_\_\_

Costo Aproximado: \_\_\_\_\_

| No. | Características de la calidad del software.   | Escala |   |   |   |   |
|-----|---|--------|---|---|---|---|
|     |   | 1      | 2 | 3 | 4 | 5 |
| 1.  | El software que utilizó realiza lo apropiado.   |        |   |   |   |   |
| 2.  | Las funciones especificadas están presentes en el software.   |        |   |   |   |   |
| 3.  | El software que utilizó genera resultados precisos o dentro de lo esperado.   |        |   |   |   |   |
| 4.  | El software que utilizó interactúa con los sistemas especificados.  |        |   |   |   |   |
| 5.  | El software que utilizó está de acuerdo con las normas y leyes vigentes.  |        |   |   |   |   |
| 6.  | El software que utilizó evita o previene el acceso no autorizado a los datos  |        |   |   |   |   |
| 7.  | El software que utilizó no presenta fallas de manera frecuente.   |        |   |   |   |   |
| 8.  | El software que utilizó reacciona adecuadamente cuando se presentan fallas.   |        |   |   |   |   |
| 9.  | El software que utilizó aun teniendo problemas mantiene determinados niveles de desempeño.                                      |        |   |   |   |   |
| 10. | El software que utilizó es capaz de recuperar datos en caso de fallas.  |        |   |   |   |   |
| 11. | El software que utilizó es capaz de restablecer el nivel de desempeño deseado.  |        |   |   |   |   |
| 12. | El software que utilizó es fácil de entender el concepto y la aplicación.   |        |   |   |   |   |
| 13. | El software que utilizó es fácil aprender a usarlo  |        |   |   |   |   |
| 14. | El software que utilizó es fácil de operar y controlar.   |        |   |   |   |   |
| 15. | El software que utilizó posee tiempo de respuesta y de ejecución adecuada   |        |   |   |   |   |
| 16. | Las tasas de procesamiento en tiempo al ejecutar las funciones prescritas son adecuadas.  |        |   |   |   |   |
| 17. | La duración del uso de los recursos (CPU, Discos, Memoria, etc.) al ejecutar las funciones prescritas es adecuada.              |        |   |   |   |   |
| 18. | Es fácil encontrar una falla cuando ocurre.   |        |   |   |   |   |
| 19. | El software que utilizó permite diagnosticar deficiencias o causas de fallas y localizar las parte modificadas para corregirlos |        |   |   |   |   |
| 20. | El software que utilizó es fácil de modificar y adaptar.  |        |   |   |   |   |
| 21. | El software que utilizó no evidencia grandes riesgos cuando se le hacen alteraciones.   |        |   |   |   |   |
| 22. | El software que utilizó es fácil de adaptar a otros ambientes.  |        |   |   |   |   |
| 23. | El software que utilizó es fácil de instalar en otros ambientes.  |        |   |   |   |   |
| 24. | El software que utilizó está de acuerdo con patrones de comportabilidad.  |        |   |   |   |   |
| 25. | El software que utilizó es fácil de usar para sustituir a otro.   |        |   |   |   |   |

Para cada dimensión el encuestado debe definir el grado de importancia(peso) que tiene esa afirmación para el en el producto de software.

Así para el encuestado "i" sus valoraciones para la D1(Funcionabilidad) estarían dado primero que todo por el peso que tiene cada una de las afirmaciones según su percepción en el sistema con valores entre 0 y 1 sumando 1 el total para esta dimensión.

Ejemplo:

Encuestado "i"

D1:

$$w_{i[1,1]}=0.2$$

$$w_{i[1,2]}=0.1$$

$$w_{i[1,3]}=0.15$$

$$w_{i[1,4]}=0.2$$

$$w_{i[1,5]}=0.3$$

$$w_{i[1,6]}=0.05$$

Cumplíendose  $w_{i[1,j]} = 1$  con  $j=0$  hasta  $j=5$ ;

Luego de definir los niveles de importancia según su percepción para cada uno de los atributos de las diferentes dimensiones comienzan los encuestados a llenar las encuestas para valorar la presencia de cada una de las características en el software evaluado.

Para medir la calidad total del software primero se pasa a una valoración de las evaluaciones en cada dimensión para esto siguiendo la teoría de las redes bayesianas se toma como la evaluación de la calidad en cada una de las dimensiones como la sumatorio del producto entre el criterio de evaluación de la característica y el peso de la misma.

Ejemplo:

Para D1 y el encuestado "i"

| Pesos                   | Evaluaciones | Producto |
|-------------------------|--------------|----------|
| $w_{i[1,1]}=0.2$<br>2   | 5            | 1        |
| $w_{i[1,2]}=0.1$<br>1   | 2            | 0.2      |
| $w_{i[1,3]}=0.15$<br>15 | 3            | 0.45     |
| $w_{i[1,4]}=0.2$<br>2   | 4            | 0.8      |



|                           |      |      |
|---------------------------|------|------|
| wi[1,5]=0.<br>3           | 3    | 0.9  |
| wi[1,6]=0.<br>05          | 5    | 0.25 |
| Nivel de Calidad<br>de D1 | 3.70 |      |

De esta manera se calculan los grado de calidad de cada una de las dimensiones, para finalmente calcular la calidad del software según el nivel de percepción del encuestado "i". Para esto se realiza un promedio de los valores de calidad de las seis dimensiones.

| Dimen<br>sión       | Nivel de<br>calidad |
|---------------------|---------------------|
| D1                  | 3.70                |
| D2                  | 4.50                |
| D3                  | 2.25                |
| D4                  | 4.90                |
| D5                  | 3.30                |
| D6                  | 4.35                |
| Calidad total: 3.83 |                     |

Lo que significa que el sistema tiene una calidad de un 76.6%, o sea el sistema evaluado posee niveles muy bajos de calidad a pesar encontrarse por encima del 60%. Se puede apreciar que la mayor incidencia sobre los bajos niveles de calidad están dado por el comportamiento de las dimensiones D3, D5 y D1, lo que indica a los diseñadores, analistas, ingenieros y programadores a prestar un especial interés en el desarrollo de estas dimensiones para próximas versiones de este sistema. Teniendo en cuenta que la metodología seguida para el desarrollo del sistema es RUP esto sirve de gran ayuda debido que en las sucesivas iteraciones de las etapas de prueba permite corregir fallas antes de la explotación del sistema.

Esta claro que hasta ahora hemos actuado con la percepción de un solo encuestado, la cual no tiene porque coincidir con todos los usuarios del sistema. Por ese motivo la encuesta es aplicada a muestras de usuarios y partir de los resultados en cada uno de los encuestados se llega a un consenso final de la calidad total del sistema.

## Bibliografía:

1. Barbacci, M (2005). Atributos de calidad de software.
2. García, Luis (2005). Calidad de SW. Parte 2-Calidad del producto(ISO 9126)
3. La Fé, Juan Alberto (2005). Procedimiento SoftQual para la medición de la calidad del software en Empresas Hoteleras.