

Universidad de las Ciencias Informáticas

Facultad 5



Trabajo de Diploma para optar por el título de:
“Ingeniero en Ciencias Informáticas”

**Título: Procedimiento para evaluar la
Mantenibilidad de componentes de software**

Autora: Camila Martínez Ortega

Tutora: MSc. Liudmila Reyes Álvarez

Co-Tutora: Ing. Yadira Morales Álamo

La Habana, 25 de mayo del 2011

“Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Facultad 5 de la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____

Camila Martínez Ortega

Firma del Autor

MSc. Liudmila Reyes Álvarez

Firma del Tutor

Ing. Yadira Morales Álamo

..

Firma del Co-Tutor

DATOS DE CONTACTO

MSc. Liudmila Reyes Álvarez

Graduada como Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2007.

Empresa: Universidad de Ciencias Informáticas UCI

Dirección: Carretera a San Antonio Km. 2 1/2 Torrens, Boyeros La Habana.

e-mail: lreyes@uci.cu Teléfono: 837-2741

Ing. Yadira Morales Álamo

Graduada como Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el año 2008. Actualmente pertenece al Grupo Calidad del Centro de Desarrollo de Informática Industrial (CEDIN) de la UCI, donde desempeña la función de Jeja del Grupo de Calidad.

Empresa: Universidad de Ciencias Informáticas UCI

Dirección: Carretera a San Antonio Km. 2 1/2 Torrens, Boyeros La Habana.

e-mail: yalamo@uci.cu Teléfono: 837- 3201

Dedico este trabajo a mi mamá por ser mi razón de ser, a mi hermanita por ser la persona más linda y angelical que siempre ha estado a mi lado. A mi abuela Rosa por siempre apoyarme y demostrarme su cariño, y a mi abuela Aydeé por todo el tiempo y amor que me dedicó.

AGRADECIMIENTO

Agradezco en primer lugar a mi mamá porque gracias a su sacrificio y apoyo he llegado hasta aquí, a mis abuelas que ofrecieron todo su amor y dedicación para mi formación.

Doy gracias a Adrián, mi compañero y novio en la universidad, que desde los inicios de mi carrera ha estado cerca de mí, ayudándome a salir adelante, a mi amiga Yanet quien ha sido como mi familia en la universidad.

Gracias a mis compañeras Nory, Dayana y Yeni, a mi compañero Ray, en particular a mis tutoras Liudmila y Yadira y a mi oponente Mariela por su colaboración en el desarrollo de este trabajo.

En fin, a esas personas que han influido en esta inolvidable etapa de mi vida.

¡Muchas Gracias!

RESUMEN

Esta investigación propone un procedimiento para la evaluación de la calidad de la mantenibilidad en el Centro de Desarrollo de Informática Industrial (CEDIN). Su realización estuvo motivada ya que el CEDIN está orientado por Líneas de Productos de Software (LPS), empleando el método Desarrollo de Software Basado en Componentes (DSBC) por su reducción en costos y tiempo, el mismo ha alcanzado madurez y ya se aplica en la industria, pero entre las deficiencias que afectan a esos componentes está la necesidad de evaluar la calidad de los mismos y se persigue con este estudio aplicar un procedimiento que permita evaluar si cumple con la calidad requerida, específicamente en cuanto a la mantenibilidad en sus productos.

Se comenzará la investigación realizando un estudio del estado del arte con vista a elaborar el marco teórico, abordando las técnicas empleadas para el mantenimiento del software, así como la aplicación de estándares con este mismo propósito. Conjuntamente se emplearán conceptos relacionados con el objetivo de estudio que facilitarán un mejor entendimiento de los aspectos necesarios a dominar para el desarrollo de la solución.

El desarrollo del procedimiento para evaluar la mantenibilidad de los componentes software está dividido en 3 etapas, en las cuales se realizan varias actividades para una mayor comprensión del mismo y se involucran artefactos tanto de entrada como de salida. La evaluación se realizará a partir de la fase de desarrollo en que se encuentre el componente, centrándose en las diferentes fases de la metodología Rational Unified Process (RUP), donde se medirán los atributos de la mantenibilidad en las fases donde ellos intervengan.

Finalmente el procedimiento será aplicado para su validación a dos componentes software desarrollados en el CEDIN y así comprobar si los mismos poseen calidad, en cuanto a la mantenibilidad.

Palabras Claves: Mantenibilidad, Calidad, Procedimiento, Evaluación

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
FUNDAMENTACIÓN TEÓRICA.....	6
1.1 CALIDAD EN LA PRODUCCIÓN DE SOFTWARE	6
1.1.1 Calidad de Software	6
1.1.2 Evaluación de la calidad del software.....	7
1.1.3 ¿Cómo se evalúa la calidad del software?	8
1.1.4 Estándares para evaluar la calidad de software.....	9
1.1.4.1 Estándar ISO/IEC 9126	9
1.1.4.2 Estándar ISO/IEC 14598	10
ISO/IEC 14598- Parte 1 Visión General.....	11
ISO/IEC 14598- Parte 2 Planificación y Gestión.....	11
ISO/IEC 14598 - Parte 3 El Proceso para Desarrolladores	11
ISO/IEC 14598 - Parte 4 El Proceso para Compradores	11
ISO/IEC 14598 - Parte 5 El Proceso para Evaluadores	11
ISO/IEC 14598 - Parte 6 Documentación de los Módulos de Evaluación	11
1.1.5 Ventajas del Modelo de Calidad del Estándar ISO/IEC 9126	12
1.2 CARACTERÍSTICA DE LA MANTENIBILIDAD	12
1.2.1 Mantenibilidad del software.....	12
1.2.2 Propiedades de la Mantenibilidad	13
1.2.3 Aspectos que influyen en la mantenibilidad.....	14
1.2.4 Medida de la Mantenibilidad	15
1.2.5 Métricas para la medición de la mantenibilidad del software.....	16
1.2.6 Relación de la mantenibilidad con el mantenimiento de software	17
1.3 TÉCNICAS PARA EL MANTENIMIENTO DEL SOFTWARE	17
1.3.1 Factores del coste del Ciclo de Vida del Software	17

1.3.2	Técnicas del Mantenimiento del Software	19
1.4	COMPONENTES SOFTWARE	19
1.4.1	¿Qué es un componente software?	19
1.4.2	¿Cómo influye la Mantenibilidad en los componentes software?	21
1.4.3	Ventajas proporcionadas por el Desarrollo de Software Basado en Componentes	21
	CONCLUSIONES PARCIALES.....	22
	PROPUESTA DEL PROCEDIMIENTO.....	23
2.1	Nombre del procedimiento.....	23
2.2	Objetivo del procedimiento	23
2.3	Alcance	23
2.4	Referencias	23
2.5	Responsables	24
2.6	Acrónimos, términos y definiciones.....	24
2.7	Normas generales.....	24
2.8	Flujo básico del procedimiento para evaluar la Mantenibilidad de Componentes Software	25
2.9	Desarrollo del procedimiento para evaluar la Mantenibilidad de Componentes Software	26
2.9.1	Etapa 1: Planificación de la Evaluación de la Mantenibilidad.....	26
2.9.1.1	Descripción de las actividades de la etapa Planificación de la Evaluación de la Mantenibilidad	27
2.9.2	Etapa 2: Descripción de la evaluación de la Mantenibilidad.....	29
2.9.2.1	Descripción de las actividades de la etapa Descripción de la evaluación de la Mantenibilidad.....	30
2.9.3	Etapa 3: Informe de los resultados de la Evaluación de la Mantenibilidad	39
2.9.3.1	Descripción de las actividades de la etapa Informe de los resultados de la Evaluación de la Mantenibilidad.	40
2.10	Descripción de las actividades del procedimiento	41
	CONCLUSIONES PARCIALES.....	42
	VALIDACIÓN DEL PROCEDIMIENTO	43
	INTRODUCCIÓN	43
3.1	Establecimiento del objetivo de la Evaluación de la Mantenibilidad	43

3.2 Definición del componente a evaluar	43
3.3 Definición del ambiente e involucrados	45
3.4 Especificación de los atributos de la mantenibilidad a evaluar en la fases del ciclo de vida.....	46
3.5 Selección de la estrategia para la evaluación en la fase del ciclo de vida	46
3.5.1 Resultados de la evaluación para el módulo HMI	46
3.5.2 Resultados de la evaluación para el módulo Seguridad	49
3.6 Establecimiento de los niveles de clasificación para las métricas empleadas en la evaluación.....	51
3.7 Categorización de los resultados obtenidos de las métricas	51
3.8 Información de los resultados	52
3.9 Representación gráfica de los resultados de la evaluación a los componentes.....	52
CONCLUSIONES PARCIALES.....	53
CONCLUSIONES GENERALES.....	54
RECOMENDACIONES.....	55
REFERENCIAS BIBLIOGRÁFICAS	56
BIBLIOGRAFÍA	58
GLOSARIO DE TÉRMINOS	60
ANEXOS	61

ÍNDICE DE FIGURAS

Ilustración 1: Relación entre las normas de las series ISO/IEC 9126 y ISO/IEC 14598.....	8
Ilustración 2: Características de Calidad.....	10
Ilustración 3: Distribución del coste del ciclo de vida	18
Ilustración 4: Etapas y actividades del procedimiento para evaluar la Mantenibilidad	25
Ilustración 5: Niveles de calificación para los resultados de las métricas	39
Ilustración 6: Descripción de las actividades del procedimiento	41
Ilustración 7: Módulos del SCADA UX	44
Ilustración 8: Interfaz del componente HMI	45
Ilustración 9: Representación gráfica de los resultados	52

ÍNDICE DE TABLAS

Tabla 1: Clasificación de los componentes software reutilizables	21
Tabla 2: Métricas Internas propuestas para evaluar los atributos de la Mantenibilidad	34
Tabla 3: Métricas Externas propuestas para evaluar los atributos de la Mantenibilidad	37
Tabla 4: Resultado de la Capacidad de análisis de fallos de las funciones de HMI	47
Tabla 5: Resultados de la Capacidad de análisis de fallos de los archivos instalados de HMI	47
Tabla 6: Resultado de la Capacidad de análisis de fallos de las dependencias de HMI	47
Tabla 7: Resultado de la Modificabilidad parametrizada de HMI	48
Tabla 8: Resultado del Cumplimiento de la mantenibilidad de HMI	48
Tabla 9: Resultado de la Capacidad de análisis de fallos de las funciones de Seguridad	49
Tabla 10: Resultado de la Capacidad de análisis de fallos de los archivos instalados de Seguridad	49
Tabla 11: Resultado de la Capacidad de análisis de fallos de los archivos instalados de Seguridad	50
Tabla 12: Resultado de la Modificabilidad parametrizada de Seguridad	50
Tabla 13: Resultado del Cumplimiento de la mantenibilidad de Seguridad	51

INTRODUCCIÓN

Debido a la progresiva y creciente producción de software en el mundo y a la contendida competencia entre sus productores se hace cada vez más necesario desarrollar software con la mayor calidad, aspecto fundamental para el reconocimiento a nivel mundial de los productos software y la satisfacción de los clientes. Los desarrolladores, apoyándose en las nuevas tecnologías, producen software cada vez más rápido debido a la competitividad del mercado y a la demanda existente. Esta precipitación en la concepción del software es una de las razones por las cuales no se cumplen todos los aspectos necesarios para producir software con la calidad requerida, ya que no se respetan los factores ni criterios establecidos por las normas de calidad implantadas.

Uno de los grandes problemas que afronta actualmente la esfera del desarrollo de soluciones informáticas es la calidad del software. Desde la década del 70, este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software, dando lugar a que surja una interrogante fundamental: ¿Cómo evaluar la calidad del software?

La Industria Cubana del Software (ICSW) no está ajena a los problemas existentes con relación a la calidad de software; uno de sus principales centros de desarrollo de soluciones informáticas es la Universidad de las Ciencias Informáticas (UCI) fundada en el 2002 con el objetivo de formar profesionales en el campo de la informática para la producción de software destinado a la informatización de la sociedad cubana y a la comercialización de sus productos software. Debido a tan primordial misión, la universidad centra todos sus esfuerzos en producir software de calidad para satisfacer las expectativas de sus clientes y alcanzar prestigio a nivel mundial, donde la competencia se incrementa cada vez más rápido; para ello se han trazado varias estrategias como es incluir el método Desarrollo de Software Basado en Componentes (DSBC), por sus ventajas en la reducción de costos, tiempo y riesgos asociados al proceso de construcción del software.

Dentro de la universidad se encuentra el Centro de Desarrollo de Informática Industrial perteneciente a la Facultad 5, el cual desarrolla software a través de Líneas de Productos de Software empleando el Desarrollo de Software Basado en Componentes como estrategia para lograr productos con mayor calidad pero incluso así los componentes aquí desarrollados aún tienen deficiencias en cuanto a esta, como así lo demuestran las evaluaciones llevadas a cabo por el Grupo de Gestión de la Calidad (GGC) del Centro.

Una de las características de calidad necesarias para obtener un producto por excelencia es la mantenibilidad, la cual brinda la posibilidad de que un sistema sea modificado para corregir fallos, mejorar su funcionamiento y adaptarse a los cambios de ambientes. En consecuencia, la mantenibilidad está relacionada con la facilidad de mantenimiento, a mayor mantenibilidad, menores costes de mantenimiento y viceversa.

Existen normas de calidad que explican los aspectos a tener en cuenta en el proceso de evaluación de la calidad como es el ejemplo de la norma ISO/IEC 9126 Ingeniería de software -Calidad del producto, la cual define seis características necesarias para la calidad del software. Otra norma aplicada en el proceso de evaluación es la ISO/IEC 14598 Tecnología de Información- Evaluación del producto software donde se define que hacer para evaluar la calidad y la norma ISO/IEC 14764 Ingeniería de software-Procesos del ciclo de vida del software -Mantenimiento que aborda sobre el proceso de mantenimiento del software después de terminado este.

Para dar solución al problema de calidad de la mantenibilidad de los componentes desarrollados en el CEDIN sería factible el uso de un procedimiento que permita a los administradores de calidad y a los desarrolladores evaluar la mantenibilidad durante el ciclo de desarrollo de los componentes a través del uso de estas normas.

Situación problemática

En el Centro de Desarrollo de Informática Industrial (CEDIN) se determina entre las principales dificultades que el grupo de calidad realiza una evaluación subjetiva de la misma, generalmente evaluando que se cumplan los requisitos solicitados por el cliente por lo que se limita a estimar otras características que influyen en la calidad o requisitos no funcionales como son la mantenibilidad, usabilidad, portabilidad y eficiencia lo que impide minimizar el equilibrio entre las características de calidad, así como ofrecer datos

respecto a puntos fuertes y débiles del producto evaluado, lo que propicia, finalmente, la baja calidad del producto. Se hace imprescindible para garantizar una óptima evaluación tener en cuenta otros atributos, o sea, evaluar objetivamente cada característica de la calidad de los componentes software. Una de las características necesarias a evaluar es la mantenibilidad la cual es precisa para localizar y reparar defectos e incluir nuevas funcionalidades, en caso de obviar esta característica provoca que los productos sean liberados inmunes a detectar deficiencias o causas de fallos, mal funcionamiento ante cambios de mejoras o adaptaciones del software a nuevos ambientes y distanciarse de las normas o convenciones que se relacionan con la mantenibilidad.

Por todo lo antes expuesto conlleva al planteamiento del **Problema Científico**:

- ✓ ¿Cómo evaluar componentes software de acuerdo a la Mantenibilidad en el Centro de Desarrollo de Informática Industrial?

Teniendo como **Objeto de Investigación**: La calidad de componentes software en cuanto a su Mantenibilidad.

Persiguiendo como **Objetivo General**: Desarrollar un procedimiento para evaluar la calidad de componentes de software en cuanto a su Mantenibilidad.

Delimitando como **Campo de Acción**: La evaluación de la calidad de componentes software en cuanto a su Mantenibilidad.

Se precisa como **Idea a defender**: Con el desarrollo de un procedimiento para evaluar la calidad del software en cuanto a su Mantenibilidad se pretende que los desarrolladores garanticen que los componentes sean mantenibles y los evaluadores evalúen objetivamente la mantenibilidad de los componentes software que se desarrollan en el Centro.

Para dar cumplimiento al objetivo general se plantean las siguientes **Tareas de Investigación**:

- ✓ Análisis de las fuentes de información para a partir del estado del arte en relación con el tema elaborar marco teórico.
- ✓ Descripción de los estándares de calidad con el objetivo de explicar el que más se adecue a la solución del procedimiento.
- ✓ Definición de conceptos de medición y evaluación de la calidad de software en cuanto a la mantenibilidad para un mejor entendimiento de la problemática.
- ✓ Estudio de la calidad de los componentes de software para definir como se podrá evaluar la calidad de un componente en cuanto a su Mantenibilidad en el Centro de Desarrollo de Informática Industrial.
- ✓ Elaboración del procedimiento para evaluar la calidad de los componentes en cuanto a la Mantenibilidad de los mismos, en el Centro de Desarrollo de Informática Industrial.
- ✓ Validación del procedimiento por el método de caso de estudio en algún componente del Centro de Desarrollo de Informática Industrial para registrar sus resultados en el Informe Evaluación de la Mantenibilidad.
- ✓ Elaboración del Informe Evaluación de la Mantenibilidad para incluirlo en el Plan de Aseguramiento de la Calidad del documento Plan de Desarrollo de Software.

Los **Métodos de la Investigación** utilizados en el presente trabajo de diploma son:

Métodos Teóricos:

- ✓ **Análisis Histórico-Lógico:** permite constatar teóricamente como ha ido evolucionando el tema en cuestión con el objetivo de caracterizar la situación actual para conocer si existe algún procedimiento que resuelva el problema científico planteado.
- ✓ **Analítico-Sintético:** permite la extracción de los elementos más importantes que se relacionan con el objeto de estudio para comprender y plantear a partir de fuentes bibliográficas la importancia de evaluar la calidad en cuanto a la mantenibilidad.

Métodos Empíricos:

- ✓ **Observación:** es el instrumento universal del científico, se realiza de forma consciente y orientada a un objetivo determinado, permite la recogida de la información en diferentes momentos de la investigación.
- ✓ **Estadístico:** permite obtener información numérica acerca de una propiedad o cualidad del objeto, donde se comparan magnitudes medibles y conocidas.

El documento está estructurado de la forma siguiente:

- ✓ En el **capítulo 1** “Fundamentación Teórica” está dedicado a la revisión del estado del arte para elaborar marco teórico donde se investiga acerca de los conceptos asociados al objeto de estudio, así como explicar el uso de las normas empleadas como guías en el desarrollo de la solución y conocer la importancia de la misma.
- ✓ En el **capítulo 2** “Propuesta del Procedimiento” se aborda acerca de cómo se desarrolla el procedimiento que facilitará a los desarrolladores concebir componentes software mantenibles y a los aseguradores de calidad evaluar que cumpla correctamente con la característica referenciada a través de estrategias de medición.
- ✓ En el **capítulo 3** “Validación del Procedimiento” se describe la validación del procedimiento tomando como caso de estudio algún componente software desarrollado por el CEDIN.
- ✓ Conclusiones de la investigación donde se exponen las principales conclusiones del trabajo presentado así como una propuesta para futuras investigaciones que se pueden derivar a partir del mismo.
- ✓ Anexos

FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

En el presente capítulo se reflejan los temas fundamentales que sustentan la investigación. En él se abordan varios conceptos asociados a la calidad del software y a la mantenibilidad. Se introduce en el tema de la evaluación de la calidad abordando a cerca de los estándares y las métricas usados en el mismo. Se hace un estudio del estado del arte para conocer sobre las técnicas empleadas en el mantenimiento del software. Se tendrá presente el dominio de estos conocimientos para el desarrollo de un nuevo procedimiento que se ajuste a los componentes software desarrollados en el CEDIN.

1.1 CALIDAD EN LA PRODUCCIÓN DE SOFTWARE

1.1.1 Calidad de Software

Según el Diccionario de la Real Academia Española (DRAE), es una propiedad o conjunto de propiedades inherentes de algo en específico, que permiten apreciarlo como igual, mejor o peor que las restantes de su especie. También es totalidad de las características de un producto o servicio que le confieren su aptitud a satisfacer las necesidades expresadas o implícitas.

De acuerdo a la terminología de la IEEE la calidad de un sistema, componente o proceso de desarrollo de software, se obtiene en función del cumplimiento de los requerimientos iniciales especificados por el cliente o usuario final. [1]

La norma ISO/IEC 9126 define la calidad del software como el grado en que un producto de software satisface las necesidades explícitamente solicitadas y las necesidades implícitas esperadas por el cliente. [2]

Según Roger S. Pressman es “la concordancia con los requisitos funcionales y de rendimiento establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. [3]

Por tanto se concluye que la calidad desde el punto de vista del software es la concordancia de este, producido con los requisitos 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. Teniendo en cuenta además que el uso de estándares y las normas de desarrollo permiten que se consiga la calidad.

1.1.2 Evaluación de la calidad del software

El Comité Técnico ISO/IEC es la fusión de la ISO (International Standards Organization) que emite normativas cuya adopción es obligatoria legalmente y la IEC (International Electrotechnical Commission) que tiene como objetivo promover la normalización en todas las cuestiones relacionadas con la electrotécnica. Debido al creciente uso de la informática, lo que conlleva que importe más la calidad que la cantidad de los productos software, estas dos organizaciones antes mencionadas desarrollaron el estándar ISO/IEC 9126/1991 “Evaluación de productos de software – Características de calidad y guías para su uso” que más tarde se dividió en dos estándares relacionados.

Esta norma definió seis características de calidad del producto software para ser evaluadas utilizando métricas validadas o comúnmente aceptadas y describió un modelo de evaluación de los productos de software. Por cuanto estas características de calidad y sus métricas asociadas pueden ser útiles no solo para la evaluación de los productos de software, sino también para definir los requisitos de calidad. Esta norma ha sido reemplazada por dos normas interrelacionadas, cada una de ellas formadas por múltiples partes: La norma ISO/IEC 9126 (Calidad de los productos de software) y la ISO/IEC 14598 (Evaluación de los productos de software).

La interrelación entre las normas ISO/IEC 9126 y las normas de la serie ISO/IEC 14598 se muestra en la Ilustración 1. [2]

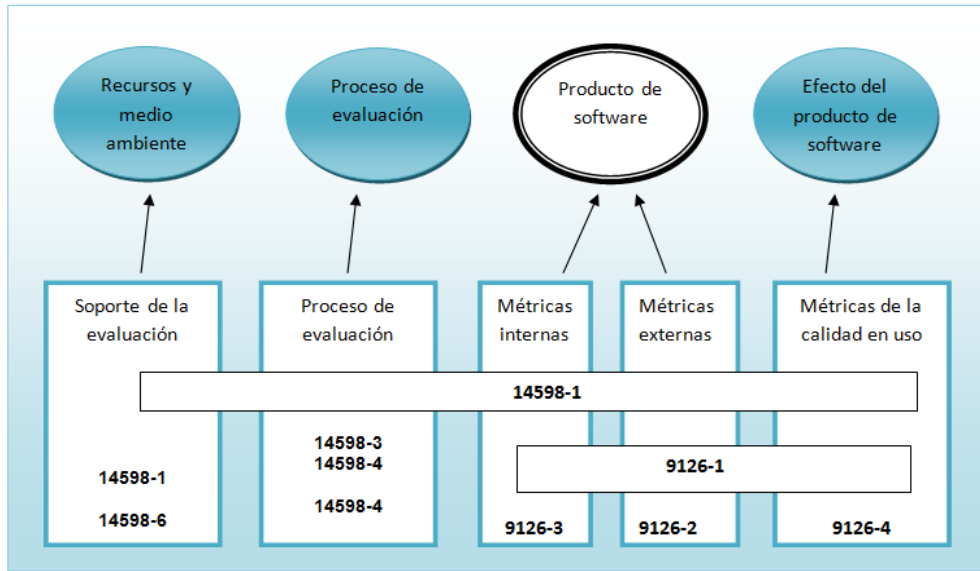


Ilustración 1: Relación entre las normas de las series ISO/IEC 9126 y ISO/IEC 14598.

La evaluación de productos del software para lograr la calidad de los mismos es uno de los procesos del ciclo de vida de desarrollo del software. La calidad del producto de software se puede evaluar al medir:

- Los atributos internos (típicamente por las medidas estáticas de productos intermedios).
- Los atributos externos (por lo general mediante el comportamiento del programa cuando se ejecuta).
- Los atributos de la calidad durante el uso (cuando el producto será usado por los clientes).

1.1.3 ¿Cómo se evalúa la calidad del software?

La evaluación de la calidad del software es el proceso de verificar que se cumpla cada característica de calidad a través de un procedimiento o técnica, guiado por estándares de calidad que facilitarán un mejor entendimiento sobre cada aspecto a evaluar.

Es prioridad que los productos software vayan siendo evaluados a medida que se van construyendo, para ello se hace necesario llevar en paralelo la evaluación al proceso de desarrollo donde participarán tanto los desarrolladores como los administradores de calidad.

Para evaluar la calidad del software es necesario, ante todo, definir una estrategia de medición. Una vez establecida esta, el proceso de evaluación, que requiere los siguientes pasos: [4]

- **Definir el software que va a ser controlado**
- **Seleccionar una medida que pueda ser aplicada al objeto de control**
- **Crear o determinar los métodos de valoración de los indicadores**
- **Definir las regulaciones organizativas para realizar el control**

1.1.4 Estándares para evaluar la calidad de software

Según la Organización Internacional para la Estandarización (ISO), un estándar es un conjunto de acuerdos documentados que contiene especificaciones técnicas u otros criterios precisos para ser usados constantemente.

1.1.4.1 Estándar ISO/IEC 9126

Durante muchos años se buscó en la Ingeniería de Software un modelo único para expresar calidad, por lo que fue propuesto como estándar internacional para la medición de la calidad del software el estándar ISO/IEC 9126, el cual establece características de calidad, subcaracterísticas y métricas asociadas que pueden utilizarse no sólo para evaluar un producto software, sino también para definir requerimientos de calidad.

Esta norma define seis características fundamentales como requisitos no funcionales que el software debe cumplir para obtener un certificado de calidad. Estas seis características son para la calidad interna y externa, que son además divididas en subcaracterísticas. Las características definidas son: [2]



Ilustración 2: Características de Calidad.

Calidad interna: total de características del producto de software desde una perspectiva interna. La calidad interna se mide y se evalúa con respecto a los requisitos de la calidad interna. Durante la aplicación del código, la revisión y el ensayo se pueden mejorar algunos detalles de la calidad del producto de software. [2]

Calidad externa: total de características del producto de software desde una perspectiva externa. Es la calidad que se obtiene cuando se ejecuta el software, y por lo general se mide y evalúa mientras se somete a ensayo en un ambiente simulado, con datos simulados, y utilizando métricas externas. [2]

Dentro de la norma ISO/IEC 9126 se incluyen los siguientes estándares: [2]

ISO/IEC 9126-1: presenta las características y subcaracterísticas para la calidad interna y externa.

ISO/IEC 9126-2: presenta las métricas de calidad externa.

ISO/IEC 9126-3: presenta las métricas de calidad interna.

ISO/IEC 9126-4: explica las métricas de calidad en el uso.

1.1.4.2 Estándar ISO/IEC 14598

La serie de normas ISO/IEC 14598 indica los requisitos a tener en cuenta para la aplicación de los métodos de medición y para el proceso de evaluación, proporcionando un entorno de trabajo para la evaluación de la calidad de diferentes tipos de productos software. Dentro de la serie ISO/IEC 14598 se pueden destacar las siguientes normas:

ISO/IEC 14598- Parte 1 Visión General

Básicamente provee una visión general de las otras cinco partes y explica la relación entre la evaluación del producto software y el modelo de calidad definido en la ISO/IEC 9126. [5]

ISO/IEC 14598- Parte 2 Planificación y Gestión

Esta parte contiene los requerimientos y las guías para las funciones de soporte tales como el planeamiento y gestión para la evaluación del producto del software. Fundamentalmente, en esta parte, se planifican las mediciones y las actividades de evaluación. [5]

ISO/IEC 14598 - Parte 3 El Proceso para Desarrolladores

Esta parte provee los requerimientos y las recomendaciones para la evaluación del producto software cuando la evaluación es conducida en paralelo con el desarrollo y llevada a cabo por el desarrollador. Se enfoca en el uso de indicadores que pueden predecir la calidad final del producto midiendo los productos intermedios que se desarrollan durante el ciclo de vida. [5]

ISO/IEC 14598 - Parte 4 El Proceso para Compradores

Esta parte provee los requerimientos y las recomendaciones para que la evaluación del producto software sea conducida en función a los compradores que planean adquirir o reutilizar un producto de software existente o pre-desarrollado. [5]

ISO/IEC 14598 - Parte 5 El Proceso para Evaluadores

Esta parte provee los requerimientos y recomendaciones para la evaluación del producto software cuando la evaluación es conducida por evaluadores independientes. [5]

ISO/IEC 14598 - Parte 6 Documentación de los Módulos de Evaluación

Esta parte provee las guías para la documentación del módulo de evaluación. Estos módulos representan la especificación del modelo de calidad y las correspondientes métricas internas y externas que serán aplicadas a una evaluación en particular. [5]

1.1.5 Ventajas del Modelo de Calidad del Estándar ISO/IEC 9126

- ✓ Las características de calidad de los productos de software definidas en esta parte de la ISO/IEC 9126 pueden ser utilizadas para especificar tanto los requisitos funcionales como no funcionales de los clientes y usuarios.
- ✓ La especificación de un modelo de calidad para atributos internos y externos.
- ✓ La coordinación del contenido con la ISO/IEC 14598.
- ✓ La ISO/IEC 9126 permite especificar y evaluar la calidad del producto de software desde las perspectivas de aquellos asociados con la adquisición, regulación, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría del software.

1.2 CARACTERÍSTICA DE LA MANTENIBILIDAD

1.2.1 Mantenibilidad del software

La IEEE (Institute of Electrical and Electronics Engineers) define la mantenibilidad como:

La facilidad con la que un sistema o componente software puede ser modificado para corregir fallos, mejorar su funcionamiento u otros atributos o adaptarse a cambios en el entorno.

Esta definición está directamente conectada con la definición del IEEE para mantenimiento del software:

Es el proceso de modificar un componente o sistema software después de su entrega para corregir fallos, mejorar su funcionamiento u otros atributos o adaptarlo a cambios en el entorno. [6]

El estándar ISO/IEC 9126 describe el concepto de mantenibilidad como la capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales. [2]

Añade que la mantenibilidad está referida por las siguientes subcaracterísticas: [2]

- **Diagnósticabilidad:** capacidad del producto del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o para identificar las partes que van a ser modificadas.
- **Flexibilidad:** capacidad del producto del software para permitir la aplicación de una modificación especificada.
- **Estabilidad:** capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas al software.
- **Contrastabilidad:** capacidad del producto del software para permitir la validación de un software modificado.
- **Conformidad de la mantenibilidad:** capacidad del producto de software para adherirse a las normas o convenciones que se relacionan con la mantenibilidad.

En consecuencia, la mantenibilidad es una característica de calidad del software relacionada con la facilidad de mantenimiento, que se considera como una actividad de mantenimiento. A mayor mantenibilidad, menores costes de mantenimiento (y viceversa).

1.2.2 Propiedades de la Mantenibilidad

La mantenibilidad se puede considerar como la combinación de dos propiedades diferentes: [7]

- Reparabilidad
- Flexibilidad

Reparabilidad

Un sistema software es reparable si permite la corrección de sus defectos con una cantidad de trabajo limitada y razonable. La reparabilidad se ve afectada por la cantidad y tamaño de los componentes o piezas. Un producto software que consiste en módulos bien diseñados es más fácil de analizar y reparar que uno monolítico, pero el incremento del número de módulos no implica un producto más reparable, ya que también aumenta la complejidad de las interconexiones entre módulos. Así pues, se debe buscar un

punto de equilibrio con la estructura de módulos más adecuada para garantizar la reparabilidad facilitando la localización y eliminación de los errores en unos pocos módulos. [7]

Flexibilidad

Un sistema software es flexible si permite cambios para que se satisfagan nuevos requerimientos, es decir, si puede evolucionar. Por su naturaleza inmaterial, es software es mucho más fácil de cambiar o incrementar por lo que respecta a sus funciones que otros productos de naturaleza física, por ejemplo, equipos hardware, pero esta flexibilidad se ve disminuida con cada nueva versión de un producto software, ya que cada versión complica la estructura del software y, por tanto, las futuras modificaciones serán más difíciles. [7]

1.2.3 Aspectos que influyen en la mantenibilidad

La mantenibilidad debe establecerse como objetivo tanto en las fases iniciales del ciclo de vida, para reducir las posteriores necesidades de mantenimiento, como durante las fases de mantenimiento, para reducir los efectos laterales y otros inconvenientes ocultos y reducir las futuras necesidades de mantenimiento.

Las técnicas de programación estructurada, la aplicación de metodologías de ingeniería del software y el seguimiento de estándares, permiten la obtención de sistemas o componentes software con menos necesidades de mantenimiento, y en el caso de que se produzcan, mucho más fáciles de llevar a cabo. Por otro lado el uso de generadores de código no produce códigos claros ni fáciles de comprender, por lo que el mantenimiento del software así generado es peor.

Aún más exacto, se han identificado los factores específicos que influyen en la mantenibilidad: [8]

- Falta de cuidado en las fases de diseño, codificación o prueba.
- Pobre configuración del producto software.
- Adecuada cualificación del equipo de desarrolladores del software.
- Estructura del software fácil de comprender.
- Facilidad de uso del sistema.
- Empleo de lenguajes de programación y sistemas operativos estandarizados.

- Estructura estandarizada de la documentación.
- Documentación disponible de los casos de prueba.
- Incorporación en el sistema de facilidades de depuración.
- Disponibilidad del equipo (computador y periféricos) adecuado para realizar el mantenimiento.
- Disponibilidad de la persona o grupo que desarrolló originalmente el software.
- Planificación del mantenimiento.

1.2.4 Medida de la Mantenibilidad

Si tenemos que elegir entre dos sistemas diferentes, ambos desarrollados con el mismo lenguaje y que tienen el mismo tamaño. La elección vendría determinada por el más fácil de mantener (lo que implica menores costes de mantenimiento), pero para saberlo tenemos que conocer su mantenibilidad.

Las medidas durante el desarrollo ayudan a determinar la cuantía en que se está incorporando en el software el objetivo de mantenibilidad. Una vez el software ha sido desarrollado, las medidas pueden guiar durante el proceso de mantenimiento, bien para evaluar el impacto de un cambio (mantenibilidad de la nueva configuración obtenida), o bien para realizar un análisis comparativo entre varias propuestas o aproximaciones diferentes para realizar una modificación requerida por los usuarios.

De los párrafos anteriores surge una pregunta: ¿Cómo medir la mantenibilidad? La respuesta a esta pregunta se encuentra en la estrecha relación que existe entre los conceptos de calidad del software y de mantenibilidad. Por esta razón, el estándar ISO/IEC 9126 propone métricas de calidad para medir la mantenibilidad del software. Las métricas de mantenibilidad no pueden medir el coste de realizar un cambio particular al sistema software, sino que miden aspectos de la complejidad y la calidad de los programas, ya que, existe una alta correlación entre la complejidad y la mantenibilidad (a mayor complejidad, menor mantenibilidad) y entre la calidad y la mantenibilidad (a mayor calidad, mayor mantenibilidad).

Existen dos aproximaciones diferentes para medir la mantenibilidad, según se consideren los aspectos externos o internos de los atributos considerados [9]. La mantenibilidad es claramente un atributo de

producto externo porque no depende únicamente del producto, sino también de la persona que realiza el mantenimiento, del soporte documental, de las herramientas disponibles, y de la utilización real del software.

La aproximación externa más directa para medir la mantenibilidad consiste en medir el proceso de mantenimiento; si el proceso es efectivo, entonces se asume que el producto es mantenible. La aproximación alternativa se utiliza para identificar atributos internos del producto y determinar cuáles de ellos son predictivos de las medidas de proceso. Aunque esta aproximación es más práctica puesto que las medidas pueden realizarse mucho más fáciles, es importante recordar que la mantenibilidad nunca se puede definir sólo en términos de medidas internas.

1.2.5 Métricas para la medición de la mantenibilidad del software

La mantenibilidad se puede medir desde el punto de vista interno a través de las métricas internas, estas se aplican al software no ejecutable, o sea en implementación, durante sus etapas de desarrollo. Las métricas internas proporcionan al usuario la capacidad de medir la calidad de los productos intermedios y por lo tanto predecir la calidad del producto final. Esto permite al usuario identificar los problemas de calidad y poner en marcha las medidas correctivas tan pronto como sea posible durante el ciclo de vida. En específico las métricas internas de mantenibilidad predicen el nivel de esfuerzo requerido por las modificaciones de los componentes software.

Si se quiere medir la mantenibilidad de los componentes software desde una perspectiva externa se aplican las métricas externas, estas se utilizan para medir la calidad del producto de software mediante la medición del comportamiento del sistema del que forma parte. Las métricas externas sólo se pueden utilizar durante las etapas de prueba del ciclo de vida. La medición se realiza al ejecutar el componente de software en el entorno del sistema en la que está destinada a funcionar. Las métricas externas de mantenibilidad persiguen el objetivo de medir el comportamiento del sistema una vez ejecutado y conocer si este puede ser modificado y así evolucionar.

1.2.5 Relación de la mantenibilidad con el mantenimiento de software

El estándar internacional ISO/IEC 14764 Ingeniería de software – Procesos del ciclo de vida del software - Mantenimiento provee una guía para el proceso de mantenimiento de software como un proceso primario en el ciclo de vida del mismo estableciendo los requisitos para llevarlo a cabo.

Este es empleado generalmente en muchos proyectos, especialmente aquellos que tienen una larga vida, ya que el mantenimiento del software es con seguridad una de las consideraciones más importantes del proyecto; donde debido al coste del producto y a las restricciones en el tiempo, además de no seguir las mejores prácticas, el software se entrega a menudo en un estado imperfecto, por tanto, es necesario ser capaces de corregir los fallos que se encuentran durante su manejo. A menudo es necesario hacer mejoras al software debido a que los requisitos y las demandas del usuario cambian. El mantenimiento del software puede llegar a ser una parte muy importante de los costes del ciclo de vida.

Las peticiones de modificación pueden ser de los siguientes tipos: [11]

- **Adaptativo:** Modificación de un producto software, después de su entrega, para conseguir que sea utilizable en un nuevo entorno.
- **Correctivo:** Modificación reactiva de un producto software, después de su entrega, para corregir defectos detectados.
- **Perfectivo:** Modificación de un producto software, después de su entrega, para mejorar su rendimiento o su mantenibilidad.
- **Preventivo:** Modificación de un producto software, después de su entrega, para detectar y corregir defectos latentes antes de que produzca fallos efectivos.

1.3 TÉCNICAS PARA EL MANTENIMIENTO DEL SOFTWARE

1.3.1 Factores del coste del Ciclo de Vida del Software

En el siguiente gráfico se muestra la distribución del coste del ciclo de vida:

Distribución del coste del ciclo de vida

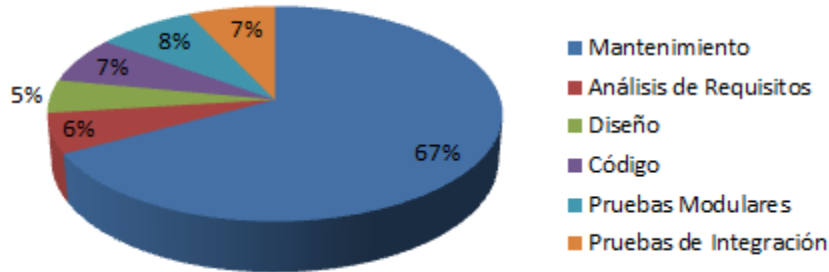


Ilustración 3: Distribución del coste del ciclo de vida.

El mantenimiento del software es una importante tarea que habitualmente requiere entre el 65% y 80% del coste del ciclo de vida del producto. Esto es debido a múltiples factores, entre los que podemos encontrar: [12]

- Inexistencia de procedimientos, técnicas y herramientas que puedan proporcionar una solución global al mantenimiento.
- La complejidad de los sistemas se incrementa paulatinamente por la realización de continuas modificaciones.
- La documentación del sistema es defectuosa e inexistente.
- Se considera el mantenimiento como una actividad poco creativa, a diferencia del desarrollo.
- Las actividades de mantenimiento se suelen realizar bajo presión de tiempo.
- Poca participación del usuario durante el desarrollo del sistema.

1.3.2 Técnicas del Mantenimiento del Software

Dentro de la ingeniería del software se proporcionan soluciones técnicas que permiten abordar el mantenimiento de manera que su impacto en coste dentro del ciclo de vida sea menor. Las soluciones técnicas pueden ser de tres tipos: [12]

1. Ingeniería inversa: Análisis de un sistema para identificar sus componentes y las relaciones entre ellos, así como para crear representaciones del sistema en otra forma o en un nivel de abstracción más elevado.

2. Reingeniería: Modificación de un producto software, o de ciertos componentes, usando para el análisis del sistema existente técnicas de ingeniería inversa y, para la etapa de reconstrucción, herramientas de ingeniería directa, de tal manera que se oriente este cambio hacia mayores niveles de facilidad en cuanto a mantenimiento, reutilización, comprensión o evolución.

3. Reestructuración del software: Cambio de representación de un producto software, pero dentro del mismo nivel de abstracción. El objetivo de estas técnicas es proporcionar métodos para reconstruir el software, ya sea reprogramándolo, redocumentándolo, rediseñándolo, o rehaciendo alguna/s característica/s del producto.

1.4 COMPONENTES SOFTWARE

1.4.1 ¿Qué es un componente software?

Los continuos avances en la informática y las telecomunicaciones están haciendo cambiar la forma en la que se desarrollan actualmente las aplicaciones software. En particular, el incesante aumento de la potencia de los ordenadores personales, el abaratamiento de los costes del hardware y las comunicaciones, y la aparición de redes de datos de cobertura global han disparado el uso de los sistemas abiertos y distribuidos.

Una solución a los problemas antes planteados ha sido la introducción del Desarrollo de Software Basado en Componentes (DSBC). Actualmente este se ha transformado en el componente más seguro tanto para la construcción de grandiosos sistemas como para aplicaciones de software. Este permite reutilizar piezas de código pre-elaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la reducción del ciclo de desarrollo y el mayor retorno sobre la inversión.

Entre las definiciones de componente software se destacan las siguientes:

Según Philippe Krutchen de Rational Rose [13], un componente es una parte no trivial, casi independiente y reemplazable de un sistema que cumple una función dentro del contexto de una arquitectura bien definida. Un componente cumple con un conjunto de interfaces y provee la realización física de ellas.

Según Clemens Szyperski [13], un componente de software es una unidad de composición con interfaces especificadas contractualmente y solamente dependencias explícitas de contexto. Un componente software puede ser desplegado independientemente y está sujeto a composición por terceros.

Sin embargo más allá de su definición existen algunas características claves para que un elemento pueda ser catalogado como componente: [14]

- ✓ **Identificable:** Debe tener una identificación que permita acceder fácilmente a sus servicios y que permita su clasificación.
- ✓ **Auto contenido:** Un componente no debe requerir de la utilización de otros para finalizar la función para la cual fue diseñado.
- ✓ **Puede ser reemplazado por otro componente:** Se puede reemplazar por nuevas versiones u otro componente que lo remplace y mejore.
- ✓ **Con acceso solamente a través de su interfaz:** Debe asegurar que estas no cambiaran a lo largo de su implementación.
- ✓ **Bien Documentado:** Un componente debe estar correctamente documentado para facilitar su búsqueda si se quiere actualizar, integrar con otros o adaptarlo.
- ✓ **Es genérico:** Sus servicios debe servir para varias aplicaciones.
- ✓ **Reutilizado dinámicamente:** Puede ser cargado en tiempo de ejecución en una aplicación.
- ✓ **Independiente de la plataforma:** Incluye hardware y software.

Existen varios tipos de componentes software reusable, entre ellos se encuentran: [15]

Según su modificabilidad <ul style="list-style-type: none">✓ Caja negra✓ Caja blanca	Según su fabricante: <ul style="list-style-type: none">✓ Componentes hechos en casa✓ COTS- Component Off the Shelf.
Según su granularidad <ul style="list-style-type: none">✓ Componentes de uso específico✓ Componentes de negocio✓ Marcos (frameworks)✓ Componentes de aplicación.	Según la tecnología usada: <ul style="list-style-type: none">✓ Componentes imperativos (Módulos, funciones)✓ Componentes OO (Clases)✓ Componentes distribuidos (Componentes CORBA, Componentes.NET, Componentes J2EE, Servicios web).

Tabla 1: Clasificación de los componentes software reutilizables.

1.4.2 ¿Cómo influye la Mantenibilidad en los componentes software?

Tras el estudio realizado acerca de la mantenibilidad se arriba a la conclusión que es imprescindible la realización de componentes software mantenibles pues es vital esta característica para estos pues su propia naturaleza los obliga a tener la capacidad de adaptarse a diferentes entornos ya que son reutilizables e igualmente necesitan evolucionar ante nuevas necesidades que surjan.

1.4.3 Ventajas proporcionadas por el Desarrollo de Software Basado en Componentes

El uso del paradigma de ensamblar componentes y escribir código para hacer que estos componentes funcionen, conocido como Desarrollo de Software Basado en Componentes posee numerosas ventajas: [16]

1. **Reutilización del software.** Nos lleva a alcanzar un mayor nivel de reutilización de software.
2. **Simplifica las pruebas.** Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
3. **Simplifica el mantenimiento del sistema.** Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.

4. **Mayor calidad.** Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

CONCLUSIONES PARCIALES

- ✓ Este capítulo ha servido para mediante elementos teóricos, afianzar científicamente la investigación, a través de definiciones, conceptos y modelos que serán el soporte en el que se apoyará el estudio que se realiza, con el fin de desarrollar el procedimiento para la evaluación de calidad de la mantenibilidad de los componentes software que se desarrollan en el CEDIN.
- ✓ Después de analizar los estándares de calidad expuestos, se decide adoptarlos como base para el desarrollo del procedimiento. Sus ventajas son las más apropiadas para realizar dicho procedimiento de evaluación haciendo mayor énfasis en la característica de mantenibilidad y sus métricas definidas en la misma, igualmente ofrece elementos importantes que son aplicables a la concepción del método de evaluación de calidad.

PROPUESTA DEL PROCEDIMIENTO

INTRODUCCIÓN

En este capítulo se explica detalladamente el procedimiento propuesto para evaluar la mantenibilidad de los productos que se desarrollan en el Centro de Desarrollo de Informática Industrial (CEDIN). Se describe la estructura del mismo a través de diferentes etapas que se involucran con las fases del ciclo de desarrollo de la metodología Rational Unified Process RUP, de igual modo se trazan actividades y se generan artefactos.

2.1 Nombre del procedimiento

Evaluación de la mantenibilidad de los componentes software desarrollados en el CEDIN:

EVAMANT-CEDIN

2.2 Objetivo del procedimiento

El procedimiento tiene como objetivo constituir una guía para el Grupo de Gestión de la Calidad del CEDIN que permita evaluar la mantenibilidad de los componentes que ahí se desarrollan, involucrando tanto los administradores de calidad como los desarrolladores, facilitando conocer el grado de mantenibilidad desde fases tempranas del ciclo de vida en el desarrollo de los componentes software e ir mitigando las deficiencias de mantenibilidad que se presenten, con el fin de lograr productos mantenibles.

2.3 Alcance

El procedimiento es aplicable a los componentes software desarrollados en el CEDIN.

2.4 Referencias

ISO/IEC 9126-1 Ingeniería de software — Calidad del producto — Parte 1: Modelo de Calidad

ISO/IEC 9126-2 Ingeniería de software — Calidad del producto — Parte 2: Métricas Externas

ISO/IEC 9126-3 Ingeniería de software — Calidad del producto — Parte 3: Métricas Internas

ISO/IEC 14598-1 Evaluación de la ingeniería de software del producto Parte 1 Aspectos Generales

ISO/IEC 14598-2 Evaluación de la ingeniería de software del producto Parte 2 Planeación y dirección.

ISO/IEC 14598-3 Evaluación de la ingeniería de software del producto Parte 3 Proceso para desarrolladores.

ISO/IEC 14598-5 Evaluación del producto de software Parte 5 Proceso para evaluadores.

IPP-1000_2008 Elaboración y aprobación de Procedimientos y Lineamientos v2.0

2.5 Responsables

Ejecuta: Administrador de Calidad y/o Desarrollador.

Responsable de su ejecución: Grupo de Gestión de la Calidad (GGC) del CEDIN.

Revisa y actualiza este procedimiento: Jefe del Grupo de Gestión de la Calidad (GGC) del CEDIN.

Fiscaliza su cumplimiento: Jefe del Grupo de Gestión de la Calidad (GGC) del CEDIN.

2.6 Acrónimos, términos y definiciones

- ✓ **Procedimiento**
- ✓ **Mantenibilidad**
- ✓ **Artefacto**
- ✓ **NC**
- ✓ **AC**

Referirse al Glosario de términos.

2.7 Normas generales

Se propone que el equipo de Aseguramiento de la calidad debe ser diferente al equipo de desarrollo del software, así se detecta con mayor facilidad los posibles errores que puedan afectar la calidad del software.

El administrador de la calidad debe poseer las competencias necesarias para la aplicación del procedimiento, tales como:

- ✓ Trabajo en equipo.
- ✓ Buen comunicador.
- ✓ Amplio poder estadístico.
- ✓ Dominio del ciclo de desarrollo del software.
- ✓ Conocimiento los principales conceptos relacionados con calidad y mantenibilidad del software.

2.8 Flujo básico del procedimiento para evaluar la Mantenibilidad de Componentes Software



Ilustración 4: Etapas y actividades del procedimiento para evaluar la Mantenibilidad.

2.9 Desarrollo del procedimiento para evaluar la Mantenibilidad de Componentes Software

El procedimiento propuesto pretende ayudar a la evaluación por el Grupo de Gestión de la Calidad del CEDIN, pues con su aplicación el proceso de detección de no conformidades se verá enriquecido, al proporcionar información para identificar problemas en la mantenibilidad de los componentes software que se desarrollan en el centro durante el proceso de desarrollo de los mismos y así lograr componentes mantenibles.

2.9.1 Etapa 1: Planificación de la Evaluación de la Mantenibilidad

En esta primera etapa **Planificación de la Evaluación de la Mantenibilidad** se lleva a cabo la planificación del trabajo, el aseguramiento de los medios tecnológicos y documentos necesarios para llevar a cabo la evaluación. De esta se sustentan las siguientes etapas, por tanto es de vital importancia que exista interacción entre miembros del proyecto que se evalúa y los evaluadores, con el objetivo de familiarizar al equipo con la característica en cuestión.

. Actividades que se desarrollan en esta etapa:

- ✓ Establecimiento del objetivo de la evaluación.
- ✓ Definición del componente a evaluar.
- ✓ Definición del ambiente e involucrados.

Artefactos que intervienen en esta etapa:

- ✓ Plan de desarrollo de software (PDS)
- ✓ Requisitos No Funcionales (RNF)
- ✓ Informe de Evaluación de la Mantenibilidad (IEM)

Roles involucrados en la etapa:

- ✓ Administrador de la Calidad
- ✓ Desarrollador

2.9.1.1 Descripción de las actividades de la etapa Planificación de la Evaluación de la Mantenibilidad

Establecimiento del objetivo de la evaluación

El propósito de la evaluación de la mantenibilidad del software es apoyar directamente al cumplimiento de los atributos de calidad que dicha característica posee: Diagnósticabilidad, Flexibilidad, Estabilidad Contrastabilidad y Conformidad de la Mantenibilidad de un componente para conocer el grado de mantenibilidad que este posee.

El propósito de la evaluación de la calidad interna de un componente puede ser:

- ✓ Decidir si el componente está siendo implementado de acuerdo a los atributos de la mantenibilidad.
- ✓ Recopilar información acerca de la evaluación de los atributos de la mantenibilidad de los componentes en implementación para controlar y gestionar la realización de los próximos procesos.
- ✓ Predecir o estimar sobre la calidad de la mantenibilidad del componente al ser desarrollado.

El propósito de la evaluación de la calidad externa de un componente puede ser:

- ✓ Decidir si el componente desarrollado es mantenible o no.
- ✓ Comparar y seleccionar un componente mantenible entre otros.
- ✓ Evaluar los efectos positivos y negativos de los atributos de la mantenibilidad de un componente para decidir cuándo mejorar o reemplazarlo.

Definición del componente a evaluar

El tipo de componente software a evaluar puede ser intermedio o final de lo que dependerá la etapa del ciclo de vida en que se encuentre y del propósito de la evaluación. Esto se refiere a que la calidad externa solo puede ser evaluada cuando el sistema de software está terminado o implementado una parte de él, o

sea un componente. Esta se lleva a cabo cuando el código está completo, en cambio, la evaluación de la calidad interna se va realizando en paralelo a la implementación del código. En caso de que la calidad externa no se logre con óptimos resultados, la evaluación puede ser utilizada como retroalimentación para modificar los atributos de mantenibilidad del software con el fin de mejorar la calidad externa, lo que apoya un proceso de mejora iterativo.

Por lo antes mencionado este procedimiento será desarrollado para facilitar el proceso de evaluación de la calidad de la mantenibilidad de los componentes software desarrollados en el CEDIN, tanto para evaluar la calidad interna por parte de los desarrolladores como la calidad externa por los administradores de calidad.

Definición del ambiente e involucrados

A continuación se definirán los recursos tanto materiales como humanos del procedimiento para la evaluación de la mantenibilidad de componentes software, se define como recurso material a todo lo que se utilizará en la ejecución del mismo y recurso humano a las personas con un rol específico que aplicarán el procedimiento que se elaborará.

Para llevar a cabo el procedimiento es preciso lograr un ambiente de trabajo propicio para su desarrollo, por lo que se hace necesario definir el lugar de la evaluación así como la cantidad de computadoras y por otra parte, es vital poseer y controlar la información con la que se trabaja en el proyecto que desarrolla el componente que se evaluará, en caso de estar ubicada en un repositorio conocer los nombres de usuario autorizados y sus contraseñas para acceder a ella, en caso de ser de escritorio contar con la versión más reciente.

En la aplicación del procedimiento es útil la conformación de un equipo con determinados conocimientos y habilidades por lo que se acogen como recursos humanos los roles de administradores de la calidad y los desarrolladores. Es importante señalar, en caso de necesidad, contar con otros roles involucrados al desarrollo del componente.

Responsabilidades del Administrador de Calidad

- ✓ Incluye en el Plan de Aseguramiento de la Calidad el procedimiento para evaluar la mantenibilidad como actividad planificada por el Grupo de Gestión de la Calidad (GGC) en la evaluación de la calidad de los componentes software.
- ✓ Agrega al Plan de Mediciones las métricas para evaluar los atributos de la mantenibilidad llevadas a cabo en la evaluación de la misma.
- ✓ Participa en el análisis y recolección de los datos en la medición de la mantenibilidad para tomar medidas ante posibles errores detectados.

Responsabilidades del Desarrollador

- ✓ Convierte la especificación del sistema en código fuente ejecutable cumpliendo con los atributos de la mantenibilidad.
- ✓ Integra los componentes que forman parte de la solución comprobando la mantenibilidad de cada uno previamente.
- ✓ Ejecuta la evaluación durante la implementación del componente y genera no conformidades asociadas a la mantenibilidad.
- ✓ Registra y analiza los resultados de la evaluación de la mantenibilidad para conocer el grado de mantenibilidad y llevar a cabo la retroalimentación en aras de corregir los errores detectados.

2.9.2 Etapa 2: Descripción de la evaluación de la Mantenibilidad

En la etapa **Descripción de la evaluación de la Mantenibilidad** se explica cómo se lleva a cabo el proceso de evaluación en las diferentes fases de la metodología Rational Unified Process (RUP), detallando los atributos de mantenibilidad que serán evaluados, a través de la estrategia definida para ello en las fases que estos intervengan. A su vez se precisa un rango para que los resultados obtenidos se puedan categorizar de acuerdo a un nivel de aceptación.

Actividades que se desarrollan en esta etapa:

- ✓ Especificación de los atributos de la mantenibilidad a evaluar en las fases del ciclo de vida de RUP.
- ✓ Selección de la estrategia para la evaluación por cada fase del ciclo de vida de RUP.
- ✓ Establecimiento de los niveles de clasificación para las métricas empleadas en la evaluación.

Artefactos que intervienen en esta etapa:

- ✓ Plan de Desarrollo de Software (PDS)
- ✓ Informe de Evaluación de la Mantenibilidad (IEM)

Roles involucrados en la etapa:

- ✓ Administrador de la Calidad
- ✓ Desarrollador

2.9.2.1 Descripción de las actividades de la etapa Descripción de la evaluación de la Mantenibilidad

Especificación de los atributos de la mantenibilidad a evaluar en las fases del ciclo de vida de RUP

La metodología Rational Unified Process (RUP) es la más apropiada para proyectos grandes (aunque también pequeños), dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas, generalmente es la más usada en la concepción de componentes software del CEDIN, en caso de que se desarrollen componentes con otra metodología el procedimiento sería similar pues generalmente mantienen las mismas fases de desarrollo. La concepción del software está particionado en diferentes ciclos, cada uno de estos trabaja en una nueva generación del componente. RUP divide un ciclo de desarrollo en cuatro fases consecutivas.

- ✓ Fase de Inicio
- ✓ Fase de Elaboración
- ✓ Fase de Construcción
- ✓ Fase de Transición

La mantenibilidad debe formar parte integral del proceso de desarrollo del software, por tal motivo es necesario en cada fase trazar una estrategia para velar por su cumplimiento, en el caso de las dos primeras fases se deben centrar las bases para el desarrollo de un componente mantenible tomando requisitos que favorezcan al cumplimiento de esta característica y para las posteriores fases evaluar los atributos relacionados con la mantenibilidad, ya sea en el código o en la ejecución del componente software, por lo tanto cuando se va a realizar la evaluación de un componente software se debe determinar si este está en proceso de elaboración o si ya está terminado.

En caso de no estar terminado se debe comenzar la evaluación en la fase del ciclo de vida en que se encuentre el componente.

➤ La Mantenibilidad en la fase de Inicio

Si el componente se encuentra en la fase de Inicio la evaluación no se realiza directamente pues en esta fase es donde se establece el caso de negocio para el sistema y delimita el alcance del proyecto. Para lograr esto se deben identificar todas las entidades externas con las cuales el sistema interactúe (los actores). Esto implica identificar todos los casos de uso y describir sólo los más significativos. El caso de negocio incluye criterios de éxito, la valoración de riesgos, la estimación de los recursos necesarios, y un plan de la fase que muestre las fechas previstas e hitos importantes.

La estrategia de evaluación de la mantenibilidad para la fase Inicio es:

- ✓ Participación de los involucrados en la definición de un componente que tenga como alcance ser mantenible.
- ✓ Estimaciones entre los involucrados de los posibles riesgos de mantenibilidad.

Muchos problemas de calidad surgen por conflictos con la mantenibilidad, para mejorarla se debe convencer a los responsables de que la mayor ganancia se obtendrá únicamente cuando la mantenibilidad esté incorporada intrínsecamente en los componentes software.

➤ La Mantenibilidad en la fase de Elaboración

Si el componente a evaluar se encuentra en la fase de Elaboración aún no se puede llevar a cabo una estrategia que evalúe la mantenibilidad del componente pero se deben ir asentando las bases para que el componente en sus futuras fases tenga éxito a la hora de medir la mantenibilidad de los mismos.

El propósito de la fase de Elaboración es analizar el dominio del problema, establecer la arquitectura, desarrollar el plan del proyecto y definir los requisitos funcionales y no funcionales del componente. En esta fase intervienen con mayor intensidad los flujos de trabajo Requisitos y Análisis y Diseño por lo que se debe definir como requisito no funcional que el componente sea mantenible lo que permitirá que el componente pueda ser modificado en un futuro y pueda evolucionar, adaptándose a nuevos requerimientos y cambios en el ambiente, característica fundamental para ser un componente reutilizable.

La estrategia de evaluación de la mantenibilidad para la fase Elaboración es:

En esta fase uno de los principales flujos de trabajo que interactúan es el de Levantamiento de Requisitos, por lo que se tiene que aplicar como estrategia el cumplimiento de los siguientes requisitos para la futura implementación del componente, estos son:

- ✓ Identificación y definición de funciones, especialmente las opcionales.
- ✓ Exactitud y organización lógica de los datos.
- ✓ Los Interfaces (de máquina y de usuario).
- ✓ Requerimientos de rendimiento.
- ✓ Requerimientos impuestos por el entorno (presupuesto).
- ✓ Granularidad (detalle) de los requerimientos y su impacto sobre la trazabilidad.
- ✓ La inclusión de los atributos de la mantenibilidad
 - Componentes software flexibles, o sea que permita cambios para que se satisfagan nuevos requerimientos, es decir, que pueda evolucionar.
 - Componente software reparables, o sea permite la corrección de sus defectos con una cantidad de trabajo limitada y razonable.
 - Componente software estables, o sea que minimice los efectos inesperados de las modificaciones realizadas.

El otro flujo de trabajo de mayor peso en esta fase es el de Análisis y Diseño donde se debe velar por el cumplimiento de los aspectos que más impactan sobre la mantenibilidad, estos son:

- ✓ Elección de la estructura del programa.
- ✓ Descomposición en módulos y flujo de datos entre ellos.
- ✓ Correcta documentación de las especificaciones de diseño, lo que conlleva a reducir los costes de mantenimiento ante modificaciones futuras, igualmente la documentación debe estar bien explícita en caso de que un componente sea transferido a un nuevo proyecto.

➤ **La Mantenibilidad en la fase de Construcción**

En la fase de Construcción el flujo de trabajo de mayor peso es la Implementación, donde el componente es desarrollado y se integran todas las subcaracterísticas para lograr un componente mantenible, por lo que se tiene que evaluar cada atributo de la mantenibilidad para prever la calidad óptima antes de las pruebas y llevar a cabo la retroalimentación para corregir los errores a tiempo.

Para que la evaluación tenga éxito y se cumpla el objetivo propuesto en esta fase, se debe ir incorporando la calidad de la mantenibilidad al proceso de desarrollo, el componente software debe cumplir con los siguientes atributos:

- ✓ Ser un componente diagnosticable.
- ✓ Ser un componente flexible.
- ✓ Ser un componente estable.
- ✓ Ser un componente contrastable.
- ✓ Ser un componente conforme a la mantenibilidad.

La estrategia de evaluación de la mantenibilidad para la fase Construcción es:

En esta fase se encuentra implementado parte del código del componente por lo que se pueden evaluar los atributos de la calidad interna, o sea cuando la evaluación se realiza al código, la cual se debe definir

CAPÍTULO 2: PROPUESTA DEL PROCEDIMIENTO

cuantitativamente mediante métricas internas establecidas para ello, donde se predice la calidad final del componente midiendo los productos intermedios que se desarrollan durante el ciclo de vida.

Dichas métricas internas se aplican al software no ejecutable, o sea en implementación. Esto permite al usuario identificar los problemas de calidad y poner en marcha las medidas correctivas tan pronto como sea posible durante el ciclo de vida.

La norma ISO/ IEC 9126-3 define una serie de métricas internas para evaluar cada uno de los atributos de la mantenibilidad durante esta fase y en este procedimiento se hace una propuesta de selección de dichas métricas para la evaluación de los mismos. Los criterios para la selección de las métricas están basados en la facilidad de entendimiento de estas, la facilidad de uso y la eficiencia para la evaluación de la subcaracterística.

Las métricas propuestas para la evaluación de la calidad interna de los atributos de la mantenibilidad que intervienen en la fase de Construcción son las siguientes:

CARACTERÍSTICA	ATRIBUTOS	MÉTRICAS INTERNAS	NIVEL REQUERIDO	PESOS (A/ M / B)
Mantenibilidad	Diagnósticabilidad	Analizabilidad (Preparación de la función de diagnóstico)	mayor de 0,5	A
	Flexibilidad	Variabilidad (Grabación de cambio)	mayor de 0,5	A
	Estabilidad	Estabilidad (Impacto de cambio)	mayor de 0,5	A
	Contrastabilidad	Capacidad de prueba (Autonomía de la capacidad de prueba)	mayor de 0,5	A
	Conformidad de la mantenibilidad	Cumplimiento de la mantenibilidad (Cumplimiento de la mantenibilidad)	mayor de 0,5	A

Tabla 2: Métricas Internas propuestas para evaluar los atributos de la Mantenibilidad.

- Para evaluar el atributo de **Diagnosticabilidad** se aplican las métricas de **Analizabilidad** las cuales se encargan de predecir el esfuerzo o recursos empleados en diagnosticar las causas de fallos o deficiencias del software, se ha seleccionado la métrica **Preparación de la función de diagnóstico** entre las métricas de Analizabilidad para evaluar este atributo ya que permite calcular la capacidad de análisis de fallas.

CAPÍTULO 2: PROPUESTA DEL PROCEDIMIENTO

- Para evaluar el atributo de **Flexibilidad** se aplica la métrica de **Variabilidad** la cual se encarga de predecir el esfuerzo de alguna modificación específica al software en este caso la métrica propuesta es **Grabación de cambio** encargada de verificar si el código esta comentareado.
- El atributo **Estabilidad** se evalúa aplicando las métricas de **Estabilidad** encargadas de predecir la estabilidad del software posterior a los cambios, para ello de selecciona la métrica **Impacto de cambio** la cual se encarga de contar los efectos adversos posteriores a las modificaciones.
- Para evaluar el atributo de **Contrastabilidad** se aplican las métricas **Capacidad de pruebas** las cuales predicen la cantidad de funciones a implementar y a través de la métrica seleccionada **Autonomía de la capacidad de prueba** se demuestra si el software puede ser capaz de probarse por sí mismo con cada función nueva que se le incorpore.
- El atributo **Conformidad de la mantenibilidad** se evalúa por la métrica del tipo **Cumplimiento de la mantenibilidad** la cual define una métrica con el mismo nombre capaz de demostrar que el software ha implementado los requisitos establecidos en el diseño para adherirse a las normas de mantenibilidad.
- El atributo **Conformidad de la mantenibilidad** se evalúa por la métrica del tipo **Cumplimiento de la mantenibilidad** la cual define una métrica con el mismo nombre capaz de demostrar que el software ha implementado los requisitos establecidos en el diseño para adherirse a las normas de mantenibilidad.

Estas métricas son incluidas en el acápite de Plan de Mediciones del Plan de Desarrollo de Software para ser tomadas en cuenta entre las métricas aplicadas al proceso de evaluación de la calidad. La descripción de cada una de las métricas internas propuestas para cada uno de los atributos de la mantenibilidad está expuesta en el Anexo1.

➤ La Mantenibilidad en la fase de Transición

Se ingresa a esta fase cuando el componente está lo suficientemente maduro para comenzar a pasar a producción. Esto requiere que un cierto subconjunto del sistema se encuentre en un nivel aceptable de la calidad de la mantenibilidad y que la documentación del usuario está disponible de modo que la transición proporcione resultados positivos para todas las partes.

La fase de Transición se centra en las actividades requeridas para poner el software en manos de los usuarios. Típicamente, esta fase incluye varias iteraciones, incluyendo lanzamientos beta, lanzamientos de disponibilidad general, así como la reparación de errores y el lanzamiento de versiones mejoradas.

Cuando el componente a evaluar se encuentra en esta fase se puede ejecutar el software y evaluar los atributos de la mantenibilidad desde una perspectiva externa, obteniéndose la calidad producto de la ejecución del software, y por lo general se mide y evalúa utilizando métricas externas. Durante el ensayo deberá ser posible detectar y eliminar la mayor parte de los defectos, sin embargo, es posible que queden algunos después del ensayo.

En esta fase los flujos de trabajo que más intervienen son Prueba y Despliegue por lo que se utilizan las métricas externas para la evaluación de la mantenibilidad. Las métricas externas de mantenibilidad miden aspectos de la complejidad y la calidad de los programas, igualmente estas métricas deben ser capaces de medir un atributo, cuando el software es objeto de mantenimiento o modificación, durante las pruebas.

La estrategia de evaluación de la mantenibilidad para la fase Transición es:

En la fase de Transición se cuenta con el componente terminado por lo que se evaluarán los atributos de la calidad externa, o sea cuando la evaluación se realiza cuando el software se está ejecutando, donde se debe medir cuantitativamente mediante métricas externas, es aquí cuando se decide la calidad del componente final. Dichas métricas externas se aplican al componente en ejecución.

Igualmente la norma ISO/ IEC 9126-2 define métricas externas para los atributos de la mantenibilidad que intervienen en la fase de Transición, de donde se extrajo la propuesta de métricas a aplicar en el

CAPÍTULO 2: PROPUESTA DEL PROCEDIMIENTO

procedimiento durante esta fase, basándose en los mismos criterios de selección anteriormente expuestos.

Las métricas propuestas para la evaluación de la calidad externa de los atributos de la mantenibilidad que intervienen en la fase de Transición son las siguientes:

CARACTERÍSTICA	ATRIBUTOS	MÉTRICAS INTERNAS	NIVEL REQUERIDO	PESOS (A/ M / B)
Mantenibilidad	Diagnósticabilidad	Analizabilidad (Capacidad de análisis de fallos)	mayor de 0,5	A
	Flexibilidad	Variabilidad (Modificabilidad parametrizada)	mayor de 0,5	A
	Conformidad de la mantenibilidad	Cumplimiento de la mantenibilidad (Cumplimiento de la mantenibilidad)	mayor de 0,5	A

Tabla 3: Métricas Externas propuestas para evaluar los atributos de la Mantenibilidad.

- El atributo **Diagnósticabilidad** se mide por la métricas de **Analizabilidad** entre ellas se propone la **Capacidad de análisis de fallos** la cual permite tener una estimación de las funciones que causan fallos.
- Para evaluar el atributo de **Flexibilidad** se definen las métricas de **Variabilidad** de las cuales se propone emplear **Modificabilidad parametrizada** la cual demuestra que se pueden hacer cambios al software partiendo de un parámetro en específico y resolver problemas.
- El atributo **Conformidad de la mantenibilidad** se evalúa a través de la métrica **Cumplimiento de la mantenibilidad** de la cual se propone aplicar la métrica que lleva su mismo nombre para tener conocimiento sobre las especificaciones del diseño que fueron implementadas y probadas y así adherirse a las convenciones de la mantenibilidad.

La norma ISO/ IEC 9126-2 establece que las métricas externas de los atributos Estabilidad (detección del comportamiento inesperado del sistema, cuando el software es objeto de pruebas) y las de Contrastabilidad (medición del comportamiento del servidor, el usuario o el propio sistema, incluyendo

CAPÍTULO 2: PROPUESTA DEL PROCEDIMIENTO

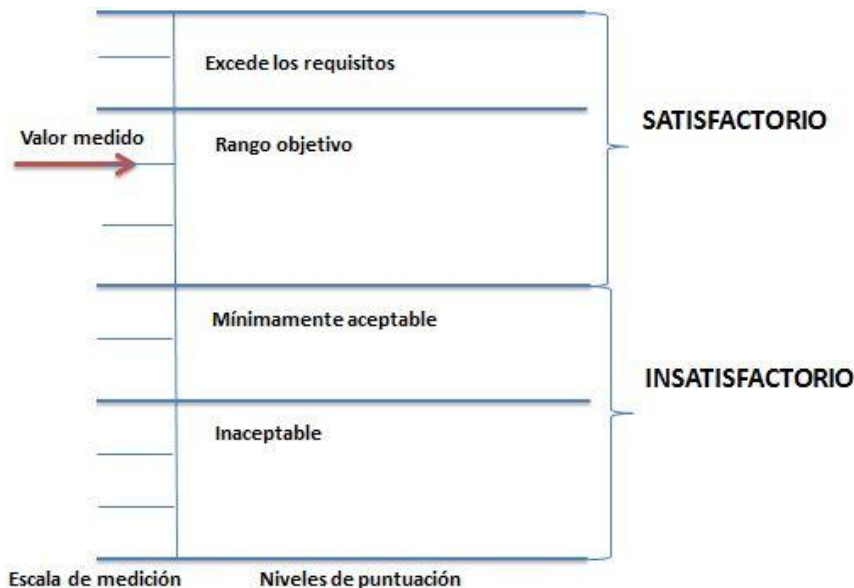
el software, cuando se intenta probar el software,) son mediciones que aún no proceden y se valoran empíricamente por las experiencias que se vayan obteniendo en la aplicación de la evaluación por tercero.

Estas métricas son incluidas en el acápite de Plan de Mediciones del Plan de Desarrollo de Software para ser tomadas en cuenta entre las métricas aplicadas al proceso de evaluación de la calidad. Cada una de las métricas externas propuestas para cada uno de los atributos de la mantenibilidad está expuesta en el Anexo2.

Es importante mencionar que posterior a la fase de Transición, se lleva a cabo una secuencia controlada y organizada de acciones para que el componente pase de ser desarrollado a ser modificado en el proceso de mantenimiento, donde los resultados de la evaluación continua de la mantenibilidad llevada a cabo en el ciclo de vida anteriormente servirá como información de entrada a este proceso y así facilitará en mismo disminuyendo sus costos.

Establecimiento de los niveles de clasificación para las métricas empleadas en la evaluación

Las características del software pueden medirse cuantitativamente utilizando métricas de calidad donde el resultado del valor de la medición se asigna a una escala dividida por dos categorías: insatisfactoria y satisfactoria, por otra parte se divide en rangos que corresponden a los diferentes grados de satisfacción de los requisitos.



Escala de medición	Niveles de puntuación	Categoría
$0,0 < V_m < 0,3$	Inaceptable	Insatisfactorio
$0,3 \leq V_m \leq 0,5$	Mínimamente aceptable	Insatisfactorio
$0,5 < V_m \leq 0,8$	Rango objetivo	Satisfactorio
$0,8 < V_m \leq 1$	Excede los requisitos	Satisfactorio

Ilustración 5: Niveles de clasificación para los resultados de las métricas.

2.9.3 Etapa 3: Informe de los resultados de la Evaluación de la Mantenibilidad

La etapa **Informe de los resultados de la Evaluación de la Mantenibilidad**, no por ser la última, es la menos importante, ya que es donde se recogen los resultados finales de la evaluación, basada en los datos arrojados por el cálculo de las métricas empleadas para la evaluación de cada atributo de la mantenibilidad y finalmente se obtiene el grado de mantenibilidad del componente como promedio de todos los resultados obtenidos para cada uno de sus atributos.

Posteriormente se registrarán los resultados obtenidos de la evaluación en el Plan de Evaluación de la Mantenibilidad.

Actividades que se desarrollan en esta etapa:

- ✓ Categorización de los resultados obtenidos de las métricas.
- ✓ Información de los resultados.

Artefactos que intervienen en esta etapa:

- ✓ Registro de No Conformidades (RNC)
- ✓ Registro de Acciones Correctivas (RAC)
- ✓ Informe de Evaluación de la Mantenibilidad (IEM)

Roles involucrados en la etapa:

- ✓ Administrador de la Calidad
- ✓ Desarrolladores

2.9.3.1 Descripción de las actividades de la etapa Informe de los resultados de la Evaluación de la Mantenibilidad

Categorización de los resultados obtenidos de las métricas

En esta actividad se comparan los resultados obtenidos de la aplicación de las métricas con los niveles de clasificación para los resultados de las mismas definidos anteriormente, con el fin de catalogarlos en satisfactorios o insatisfactorios.

Información de los resultados

En esta actividad se realiza un Informe de Evaluación de la Mantenibilidad donde se registran los resultados de la evaluación de la mantenibilidad, los cuales se mostrarán a la Alta Gerencia y se incluyen en el acápite Plan de Aseguramiento de la Calidad del Plan de Desarrollo de Software.

2.10 Descripción de las actividades del procedimiento

Procedimiento para evaluar la mantenibilidad de componentes software del CEDIN (EVAMANT-CEDIN)			
Rol Involucrado	Artefactos de entrada	Actividades	Artefactos de salida
		Inicio	
Administrador de la Calidad Desarrollador	Plan de Desarrollo de Software	Establecimiento del objetivo	Informe de Evaluación de la Mantenibilidad (llenado)
Administrador de la Calidad Desarrollador	Plan de Desarrollo de Software	Definición del componente	Informe de Evaluación de la Mantenibilidad (llenado)
Administrador de la Calidad Desarrollador	Plan de Desarrollo de Software	Definición del ambiente e involucrados	Informe de Evaluación de la Mantenibilidad (llenado)
Administrador de la Calidad Desarrollador	Requisitos No Funcionales	Especificación de los atributos de la mantenibilidad a evaluar	Informe de Evaluación de la Mantenibilidad (llenado)
Administrador de la Calidad Desarrollador	Plan de Desarrollo de Software	Selección de la estrategia de evaluación para cada fase de RUP	Informe de Evaluación de la Mantenibilidad (llenado) Registro de No Conformidades (llenado)
Administrador de la Calidad Desarrollador	Informe de Evaluación de la Mantenibilidad	Establecimiento de los niveles de clasificación para los resultados	Plan de Evaluación de la Mantenibilidad (llenado)
Administrador de la Calidad Desarrollador	Informe de Evaluación de la Mantenibilidad Registro de No Conformidades	Categorización de los resultados	Plan de Evaluación de la Mantenibilidad (llenado)
Administrador de la Calidad Desarrollador	Plan de Evaluación de la Mantenibilidad	Información de los resultados	Informe de Evaluación de la Mantenibilidad (llenado) Registro de Acciones Correctivas (llenado)
		Fin	

Ilustración 6: Descripción de las actividades del procedimiento.

CONCLUSIONES PARCIALES

- ✓ En este capítulo se presentó una propuesta del procedimiento para evaluar la mantenibilidad de los componentes software desarrollados en el CEDIN en las diferentes fases que se puedan encontrar en el ciclo de vida del mismo, lo que permitirá la planeación desde el inicio de desarrollar un componente mantenible y a su vez ir mitigando las dificultades referentes a la mantenibilidad por cada fase.

VALIDACIÓN DEL PROCEDIMIENTO

INTRODUCCIÓN

El objetivo principal de este capítulo es obtener valores reales acerca de la propuesta del procedimiento planteado, basado en el cálculo de las métricas seleccionadas para evaluar los atributos de mantenibilidad como estrategia de evaluación. A partir de la validación del procedimiento se tomarán en cuenta los resultados obtenidos por la evaluación de la mantenibilidad para identificar No Conformidades al componente evaluado para posteriormente ser corregidas y lograr un producto con mayor calidad en cuanto a ese aspecto.

3.1 Establecimiento del objetivo de la Evaluación de la Mantenibilidad

Se persigue como objetivo para la siguiente evaluación comprobar si los componentes HMI y Seguridad una vez terminados poseen calidad en cuanto a su mantenibilidad para ser utilizados con éxito en productos software, o sea conocer si son componentes mantenibles o no.

3.2 Definición del componente a evaluar

Dentro del Centro de Informática Industrial (CEDIN), se encuentran varios proyectos, uno de ellos es Energía-UCI del SCADA UX encargado de supervisar y monitorear el comportamiento eléctrico de diferentes edificaciones de la universidad; actualmente tiene un alcance de 18 locales entre los que se encuentran 6 docentes, los 3 complejos de comedores, el hospital, departamento de SOFTEL, el nodo central, el rectorado, la biblioteca, la panadería, la pizzería, la casona y el departamento de Servicios Técnicos.

El proyecto Energía-UCI está compuesto por componentes del SCADA UX, dos de ellos son HMI (Human Machine Interface) y Seguridad, ambos componentes son del tipo Módulo ya que la aplicación está ensamblada por módulos software reutilizables que han sido diseñados previamente con independencia de las aplicaciones en las que van a ser utilizados. La clasificación de componentes de tipo Módulo entra dentro de los tipos de componentes clasificados Según la Tecnología Usada ya que emplean tecnologías que permiten la integración de módulos en la arquitectura del software. A su vez se aplica la programación

orientada a objeto ya que el desarrollo basado en componentes es una evolución natural de la misma donde se aplica el encapsulamiento y distintas clases de polimorfismo.

Estos dos componentes se encuentran en la fase de Transición ya que ambos están en el proceso de Despliegue.



Ilustración 7: Módulos del SCADA UX.

El componente HMI (Human Machine Interface) del SCADA UX constituye una interfaz que permite la supervisión y control de forma gráfica de los procesos y el sistema, desde cualquier PC cliente se puede acceder a los datos que se encuentran en el servidor, este comprende todos los puntos de contacto entre el usuario y el equipo, de forma amena. La calidad del funcionamiento del componente depende en gran medida del abarrotamiento y velocidad de la red.

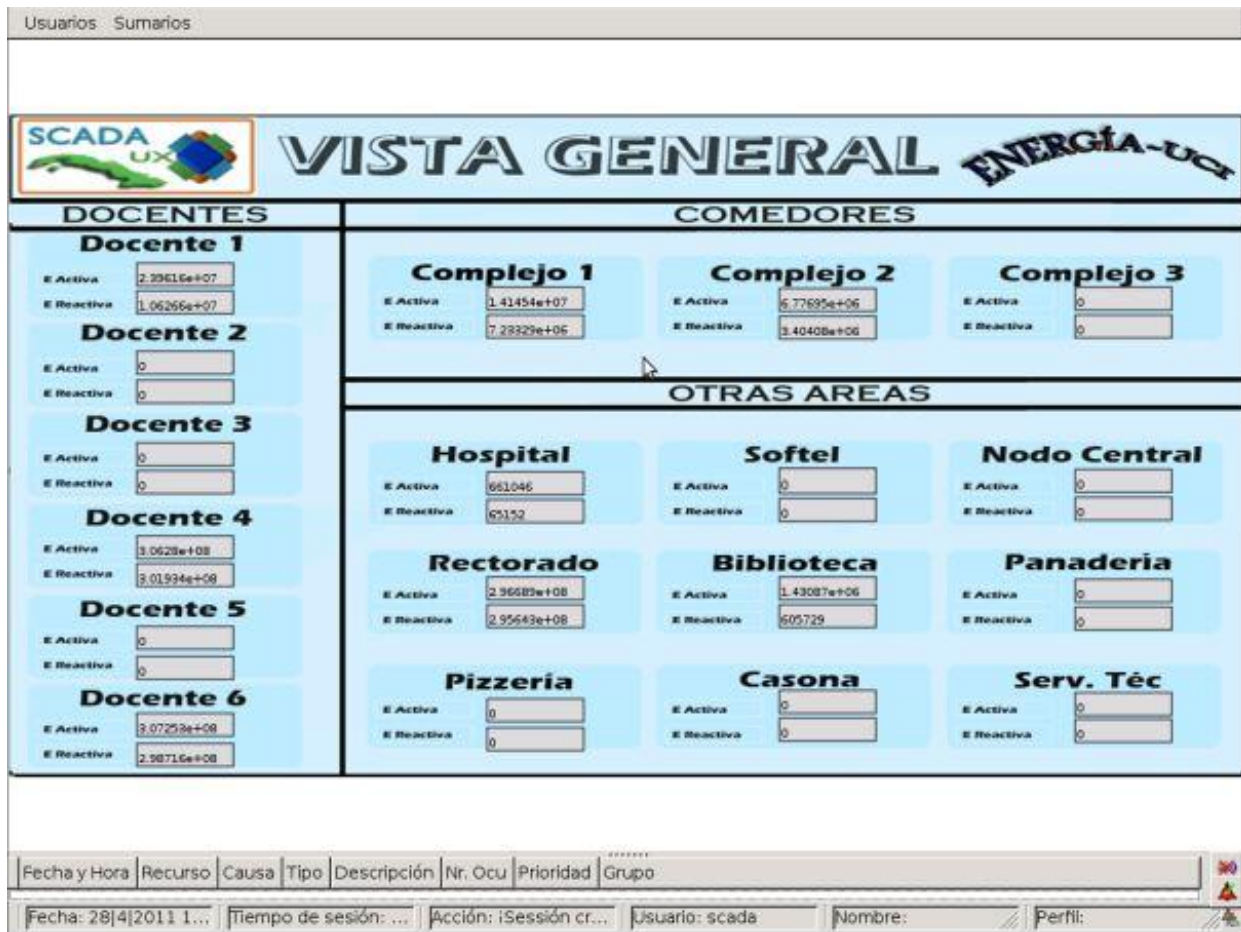


Ilustración 8: Interfaz del componente HMI.

El componente Seguridad del SCADA UX proporciona las funcionalidades necesarias para garantizar el trabajo autorizado a usuarios, además brinda las herramientas necesarias para la protección contra ataques maliciosos o involuntarios al sistema por parte de personas o recursos.

3.3 Definición del ambiente e involucrados

Se propició un ambiente adecuado para la evaluación, instalándose en una computadora el servidor del SCADA UX y posteriormente se ejecutaron los componentes a evaluar. La computadora utilizada tenía características adecuadas como la instalación del sistema operativo Debian Lenny con una memoria RAM de 1 GB.

La evaluación se llevó a cabo por parte del Administrador de la Calidad, y la ayuda del mantenedor del sistema ya que proporcionó información relacionada con el tema.

3.4 Especificación de los atributos de la mantenibilidad a evaluar en la fases del ciclo de vida

Como ambos componentes, tanto HMI como Seguridad del proyecto SCADA UX Energía-UCI se encuentran en fase de Transición se les medirán a ambos los atributos de la mantenibilidad que intervienen en esa fase, como está definido en el capítulo anterior.

Ellos son:

- Diagnósticabilidad
- Flexibilidad
- Conformidad de la Mantenibilidad

3.5 Selección de la estrategia para la evaluación en la fase del ciclo de vida

La estrategia propuesta para evaluar los atributos de la mantenibilidad en la fase de Transición es la aplicación de métricas externas capaces de medir la presencia y el cumplimiento de los atributos evaluados en el componente software y concluir midiendo el grado de mantenibilidad que estos poseen.

Las métricas externas para evaluar la los atributos de la mantenibilidad de ambos componentes demostró los siguientes resultados:

3.5.1 Resultados de la evaluación para el módulo HMI

- **Atributo Diagnósticabilidad**

Para evaluar el atributo de Diagnósticabilidad en el módulo HMI se empleó la métrica **Capacidad de análisis de fallos** para detectar los fallos de las funciones, archivos instalados y dependencias. Ver Anexo 2.

CAPÍTULO 3: VALIDACIÓN DEL PROCEDIMIENTO

Este fue el resultado de la métrica en la detección de fallos en las funciones.

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Diagnósticabilidad	Capacidad de análisis de fallos	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 1/5$	0,8	Satisfactorio

Tabla 4: Resultado de la Capacidad de análisis de fallos de las funciones de HMI.

Este fue el resultado de la métrica en la detección de fallos en los archivos instalados.

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Diagnósticabilidad	Capacidad de análisis de fallos	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 5: Resultados de la Capacidad de análisis de fallos de los archivos instalados de HMI.

Este fue el resultado de la métrica en la detección de fallos en las dependencias.

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Diagnósticabilidad	Capacidad de análisis de fallos	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 6: Resultado de la Capacidad de análisis de fallos de las dependencias de HMI.

CAPÍTULO 3: VALIDACIÓN DEL PROCEDIMIENTO

Promediando estos 3 valores se llega a la conclusión que el valor final de la métrica es de 0,9 por tanto el componente HMI es Diagnosticable.

➤ Atributo Flexibilidad

Para evaluar el atributo de Flexibilidad en el módulo HMI se empleó la métrica **Modificabilidad parametrizada** la cual demuestra si se puede modificar el software a través de un parámetro en específico. Ver Anexo 2.

Para lo cual se obtuvo el siguiente resultado:

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Flexibilidad	Modificabilidad parametrizada	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 7: Resultado de la Modificabilidad parametrizada de HMI.

➤ Atributo Conformidad de la mantenibilidad

Para evaluar el atributo de Conformidad de la mantenibilidad en el módulo HMI se empleó la métrica **Cumplimiento de la mantenibilidad** la cual demuestra si se implementaron y se probaron los requisitos especificados en el diseño. Ver Anexo 2.

Para lo cual se obtuvo el siguiente resultado:

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Conformidad de la mantenibilidad	Cumplimiento de la mantenibilidad	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 8: Resultado del Cumplimiento de la mantenibilidad de HMI.

3.5.2 Resultados de la evaluación para el módulo Seguridad

➤ **Atributo Diagnosticabilidad**

Al igual que en el módulo HMI, el atributo de Diagnosticabilidad en el módulo Seguridad se empleó la métrica **Capacidad de análisis de fallos** para detectar los fallos de las funciones, archivos instalados y dependencias. Ver Anexo 2.

Para este componente se obtuvieron los siguientes resultados:

Este fue el resultado de la métrica en la detección de fallos en las funciones.

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Diagnosticabilidad	Capacidad de análisis de fallos	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 9: Resultado de la Capacidad de análisis de fallos de las funciones de Seguridad.

Este fue el resultado de la métrica en la detección de fallos en los archivos instalados.

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Diagnosticabilidad	Capacidad de análisis de fallos	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 10: Resultado de la Capacidad de análisis de fallos de los archivos instalados de Seguridad.

CAPÍTULO 3: VALIDACIÓN DEL PROCEDIMIENTO

Este fue el resultado de la métrica en la detección de fallos en las dependencias.

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Diagnósticabilidad	Capacidad de análisis de fallos	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 11: Resultado de la Capacidad de análisis de fallos de las dependencias de Seguridad.

Promediando estos 3 valores se llega a la conclusión que el valor final de la métrica es de 1,0 por tanto el componente Seguridad es Diagnosticable.

➤ Atributo Flexibilidad

Para evaluar el atributo de Flexibilidad en el módulo Seguridad se empleó la métrica **Modificabilidad parametrizada** la cual demuestra si se puede modificar el software a través de un parámetro en específico. Ver Anexo 2.

Para lo cual se obtuvo el siguiente resultado:

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Flexibilidad	Modificabilidad parametrizada	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 12: Resultado de la Modificabilidad parametrizada de Seguridad.

➤ Atributo Conformidad de la mantenibilidad

Para evaluar el atributo de Conformidad de la mantenibilidad en el módulo Seguridad se empleó la métrica **Cumplimiento de la mantenibilidad** la cual demuestra si se implementaron y se probaron los requisitos especificados en el diseño. Ver Anexo 2.

Para lo cual se obtuvo el siguiente resultado:

Atributo	Métrica	Nivel Requerido	Fórmula	Resultados	
				Valor Métrica	Categoría
Conformidad de la mantenibilidad	Cumplimiento de la mantenibilidad	Mayor de 0,5	$X = 1 - A/B$ para $X = 1 - 0$	1,0	Satisfactorio

Tabla 13: Resultado del Cumplimiento de la mantenibilidad de Seguridad.

3.6 Establecimiento de los niveles de clasificación para las métricas empleadas en la evaluación

El resultado de cada una de estas métricas será un valor entre 0.0 y 1.0 a medida que el valor obtenido se acerque más a 1.0 tendrá mayor beneficio para la mantenibilidad, se promediará el resultado obtenido por la métrica en cada atributo para al final ofrecer si es satisfactorio o insatisfactorio, basándose en la clasificación propuesta en el capítulo anterior.

3.7 Categorización de los resultados obtenidos de las métricas

Una vez terminado el proceso de medición se obtuvo que los atributos del componente HMI obtuvieron la siguiente clasificación:

- Diagnósticabilidad con un valor de 0,9 por tanto el resultado es satisfactorio.
- Flexibilidad con un valor de 1,0 por tanto el resultado es satisfactorio.
- Conformidad con la Mantenibilidad con un valor de 1,0 por tanto el resultado es satisfactorio.

De igual modo en el componente Seguridad se obtuvieron los siguientes resultados:

- Diagnosticabilidad con un valor de 1,0 por tanto el resultado es satisfactorio.
- Flexibilidad con un valor de 1,0 por tanto el resultado es satisfactorio.
- Conformidad con la Mantenibilidad con un valor de 1,0 por tanto el resultado es satisfactorio.

3.8 Información de los resultados

La validación del procedimiento para evaluar la mantenibilidad de componentes software aplicado a los módulos HMI y Seguridad del proyecto Energía-UCI fue llevada a cabo el 28 de abril de 2011, arrojando valores positivos para ambos componentes, por tanto los dos módulos tienen calidad en cuanto a la mantenibilidad lo que permitirá realizarle posteriores modificaciones a los componentes y así tener facilidades para evolucionar. Como conclusión final los componentes HMI y Seguridad utilizados en el proyecto Energía-UCI son mantenibles obteniéndose un grado de mantenibilidad de 0,9 y 1,0 respectivamente.

En esta evaluación no se encontraron No Conformidades por tanto no se llevan a cabo Acciones Correctivas ya que la dificultad encontrada con las funcionalidades del componente HMI no dependen de la implementación del sistema sino de factores externos al proyecto.

3.9 Representación gráfica de los resultados de la evaluación a los componentes

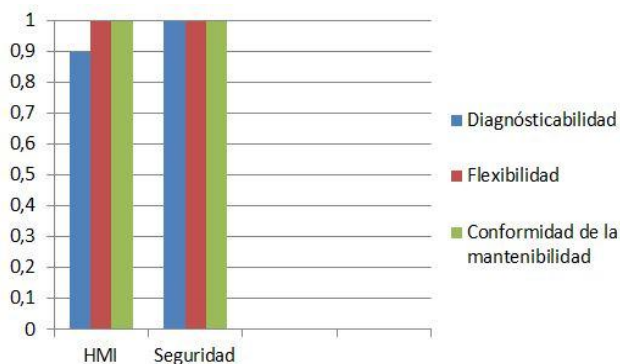


Ilustración 9: Representación gráfica de los resultados.

CONCLUSIONES PARCIALES

- ✓ En este capítulo se presentó la validación del procedimiento propuesto el cual contribuye en el conocimiento del grado de mantenibilidad de los componentes evaluados detectando los atributos de la mantenibilidad que se corresponden a él.

CONCLUSIONES GENERALES

En la aplicación de este procedimiento se le da cumplimiento al objetivo por el cual se llevó a cabo esta investigación por lo que se concluye de la siguiente forma:

- Se analizaron las fuentes bibliográficas en relación con la calidad y la evaluación de la mantenibilidad del software, así como normas que abordan el proceso de evaluación, lo cual permitió determinar los elementos necesarios para definir el procedimiento.
- Se propuso un procedimiento para evaluar la mantenibilidad de los componentes software desarrollados en el CEDIN el cual se debe incluir como una actividad más del Grupo de Gestión de la Calidad.
- Se definieron roles, responsabilidades y actividades para la evaluación, así como los artefactos que se utilizarían para la recogida de información y la salida de los resultados.
- Se validó la propuesta a través de su aplicación a dos componentes desarrollados en el CEDIN, con el objetivo de comprobar si el mismo soluciona los problemas existentes.

RECOMENDACIONES

Se recomienda:

- Realizar una capacitación a todo el personal del Grupo de Gestión de la Calidad del Centro y a los roles involucrados en el proceso de desarrollo de los proyectos encaminados al desarrollo de software basado en componentes para así desde las primeras fases del ciclo de vida del software se tengan en cuenta los aspectos relacionados con la mantenibilidad, con el objetivo de efectuar una mejor evaluación y así obtener mejores resultados.
- Profundizar en la aplicación de técnicas y herramientas que faciliten el proceso de mantenimiento y aplicarlas para reducir los costos y tiempo del mismo.
- Automatizar el proceso de evaluación a través de las métricas en una herramienta que permita guardar los resultados.
- Incorporar al procedimiento otras actividades que permitan evaluar otras características de calidad definidas en el estándar ISO/IEC 9126.

REFERENCIAS BIBLIOGRÁFICAS

- [1] **Dávila Nicanor, Leticia y Mejía Álvarez, Pedro.** *Evaluación de la Calidad de Software en Sistemas de Información en Internet.* Zacatenco, México. [En línea] <http://delta.cs.cinvestav.mx/~pmejia/davila-mejia.pdf>
- [2] *Ingeniería de software-Calidad del producto-Parte1: Modelo de la Calidad (ISO/IEC 9126-1:2001, IDT).*
- [3] **Pressman, Roger S.** *Libro Ingeniería de Software. Un enfoque práctico.* 1998.
- [5] **Aramayo, Roger Saravia.** *Postgrado en informática. Evaluación del producto software iso 14598.* noviembre de 2007. [En línea] <http://es.scribd.com/doc/14897492/Estandar-ISOIEC14598-Evaluacion-del-Producto-Software>
- [6] **Sicilia, Miguel-Ángel.** Definición de Mantenibilidad Noviembre 7, 2008. *Connexions.* [En línea] <http://cnx.org/content/m17457/1.2/>.
- [7] **Sicilia, Miguel-Ángel.** Propiedades de la Mantenibilidad Noviembre 30, 2008. *Connexions.* [En línea] <http://cnx.org/content/m17471/1.2/>.
- [8] **Francisco Ruiz, Macario Polo.** Mantenimiento del Software . *Grupo Alarcos.* Ciudad Real, 2000/2001. [En línea] <http://alarcos.inf-cr.uclm.es/doc/mso/> .
- [9] **Francisco Ruiz, Macario Polo.** *Master en Ingeniería del Software Mantenimiento del Software .* 2007.
- [10] *Technical Report ISO/IEC 9126-2 Software engineering –Product Quality Part 2: External Metrics.*
- [11] **Francisco Ruiz, Macario Polo.** Mantenimiento del software. *Grupo Alarcos. Escuela Superior de Informática* Universidad de Castilla- La Mancha [En línea] <http://alarcos.inf-cr.uclm.es/doc/mso/>
- [12] **Sicilia, Miguel-Ángel.** Técnicas de Mantenimiento de Software Febrero 5, 2011. *Connexions.* [En línea] <http://cnx.org/content/col10571/1.6>
- [13] **Montilva Jonás, Arapé Nelson , Colmenares Juan Andrés.** Desarrollo de Software Basado en Componentes Universidad de Los Andes Facultad de Ingeniería *Escuela de Ingeniería de Sistemas Departamento de Computación*

[14] **Rojas Maribel, Molina Juan Carlos** Introducción y principios básicos del desarrollo del software basado en componentes Septiembre 30 de 2004.

[15] Ingeniería de dominio . [En línea] <http://www.slideshare.net/viri88/ingenieria-de-dominio> .

[16] **Terreros, Julio Casal.** Desarrollo de Software basado en Componentes. [En línea] <http://msdn.microsoft.com/es-es/library/bb972268.aspx>.

BIBLIOGRAFÍA

Cueva, Juan Manuel. 1999. *Calidad del software*. s.l. : Departamento de Informática de la Universidad de Oviedo, 1999.

Donadello, Domingo F. 2008. *Proceso de Certificación y ejemplos de un proceso de desarrollo de software con ISO/IEC 90003 y 14598*. 2008.

Evelyn Mulén, Yoan López. 2010. *Procedimiento para la evaluación de la usabilidad de los productos software desarrollados en el CESIM* . Ciudad de La Habana : s.n., 2010.

Fillottrani, Pablo F. 2007. *Calidad en el Desarrollo de Software Modelos de Calidad de Software*. s.l. : Universidad Nacional del Sur, 2007.

García, Oscar Gómez. 2009. *SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad*. . s.l. : Universidad de Castilla-La Mancha, 2009.

2004. *GUÍA TÉCNICA SOBRE EVALUACIÓN DE SOFTWARE PARA LA ADMINISTRACIÓN PÚBLICA*. 2004.

Natalia Juristo, Ana M Moreno, Sira Vegas. 2006. *TÉCNICAS DE EVALUACIÓN DE SOFTWARE* . 2006.

Sicilia, Miguel-Angel. 2008. *Tipos de Mantenimiento*. 2008.

Silva, João Manuel Días. *FORMULACIÓN DE UN PROCEDIMIENTO PARA LA EVALUACIÓN Y CONTROL DE LA CALIDAD DE LAS INNOVACIONES RELACIONADAS CON LA ELABORACIÓN DE SOFTWARE*. San Vicente, República de Cabo Verde : Instituto de Enseñanz Superior Isidoro da Grada.

Technical Report ISO/IEC 9126-3 Software engineering –Product Quality Part 3: Internal Metrics.

International Standard ISO/IEC 14598-1:1999 Information Technology –Software Product Evaluation Part 1: General Overview.

International Standard ISO/IEC 14598-2:2000 Information Technology –Software Product Evaluation Part 2: Planning and management.

International Standard ISO/IEC 14598-3:2000 Information Technology –Software Product Evaluation Part 3: Process for developer.

International Standard ISO/IEC 14598-5:1998 Information Technology –Software Product Evaluation Part 5: Process for evaluators.

Lamaysi, Samira. 1999. *La norma ISO 14764.* 1999.

GLOSARIO DE TÉRMINOS

- ✓ **Procedimiento:** forma especificada para llevar a cabo un tipo de actividad para alcanzar un fin. Un procedimiento deberá proporcionar información clara, concisa y completa; expresada de forma sencilla y secuencial, aunque no pobre, ni monótona sobre qué cosa hacer, cómo hacerla, cuándo hacerla, cuánto hacer, dónde hacerla y quien hará y/o se responsabilizará de lo hecho.
- ✓ **Mantenibilidad:** característica (requisito no funcional) de calidad que debe poseer el software, permitiendo que este sea capaz de ser modificado para evolucionar ante la inclusión de nuevos requerimientos o necesidades de cambio de ambiente.
- ✓ **Artefacto:** productos tangibles del proyecto, que son producidos, modificados y usados para realizar actividades.
- ✓ **NC:** No conformidades o sea incumplimiento de algún requisito.
- ✓ **AC:** Acciones Correctivas o sea acción tomada para eliminar la causa de una no conformidad, defecto o cualquier situación indeseable existente para evitar su repetición.

ANEXOS

Anexo1: Métricas Internas propuestas

Métrica de Diagnosticabilidad	
Nombre de la métrica	Preparación de la función de diagnóstico
La métrica se propone medir	¿Cómo se completa la prestación de la función de diagnóstico?
Método de aplicación	Cuenta el número de funciones de diagnóstico tal como se especifica y compara con el número de funciones de diagnóstico requeridas en las especificaciones.
Medición (fórmula)	$X = A / B$ <p>A= número de funciones de diagnóstico tal como se especifica confirmado en la revisión.</p> <p>B = número de funciones de diagnóstico necesarias.</p>
Interpretación del valor obtenido	$0 \leq X$ La ejecución más cercana a 1, la mejor de las funciones de diagnóstico.

Métrica de Variabilidad	
Nombre de la métrica	Grabación de cambio
	¿Están los cambios en las especificaciones y los módulos del programa registrado adecuados a las líneas de comentarios en el código?
Método de aplicación	Proporción del registro de información de cambio de módulo
Medición (fórmula)	$X = A/B$ A = número de cambios en la función de los módulos con un cambio de comentario confirmado en la revisión. B = número total de la función de los módulos que ha cambiado desde el código original
Interpretación del valor obtenido	$0 \leq X \leq 1$ Cuanto más se acerque a 1, mayor grabación.

Métrica de Estabilidad	
Nombre de la métrica	Impacto de cambio
La métrica se propone medir	¿Cuál es la frecuencia de efectos adversos después de la modificación?
Método de aplicación	Cuente el número de efectos adversos a detectar después de la modificación y la comparemos con el número de modificaciones realizadas.
Medición (fórmula)	$X = 1 - A/B$ A = número de efectos adversos a detectar después de las modificación. B = número de modificaciones realizadas.
Interpretación del valor obtenido	$0 \leq X \leq 1$ Cuanto más se acerque a 1, mejor.

Métrica de Capacidad de prueba	
Nombre de la métrica	Autonomía de la capacidad de prueba
La métrica se propone medir	¿Cómo puede el software de forma independiente ser probado?
Método de aplicación	Cuente el número de dependencias en otros sistemas de pruebas que se han simulado con colillas y compararlo con el número total de las dependencias de la prueba en otros sistemas.
Medición (fórmula)	$X = A/B$ A = número de dependencias en otros sistemas de pruebas que se han simulado con colillas. B = número total de las dependencias de la prueba en otros sistemas.
Interpretación del valor obtenido	$0 \leq X \leq 1$ Cuanto más se acerque a 1, mejor.

Métrica de Cumplimiento de la mantenibilidad	
Nombre de la métrica	Cumplimiento de la mantenibilidad
La métrica se propone medir	¿Cómo es compatible el mantenimiento del producto con las normas aplicables, las normas y convenciones?
Método de aplicación	Cuente el número de requisitos que requieren de cumplimiento que se han cumplido y comparar con el número de requisitos que requieran el cumplimiento.
Medición (fórmula)	$X = A/B$ A = número de requisitos correctamente relacionados con el cumplimiento de mantenimiento. B = número total de elementos de cumplimiento.
Interpretación del valor obtenido	$0 \leq X \leq 1$ Cuanto más se acerque a 1, el más dócil.

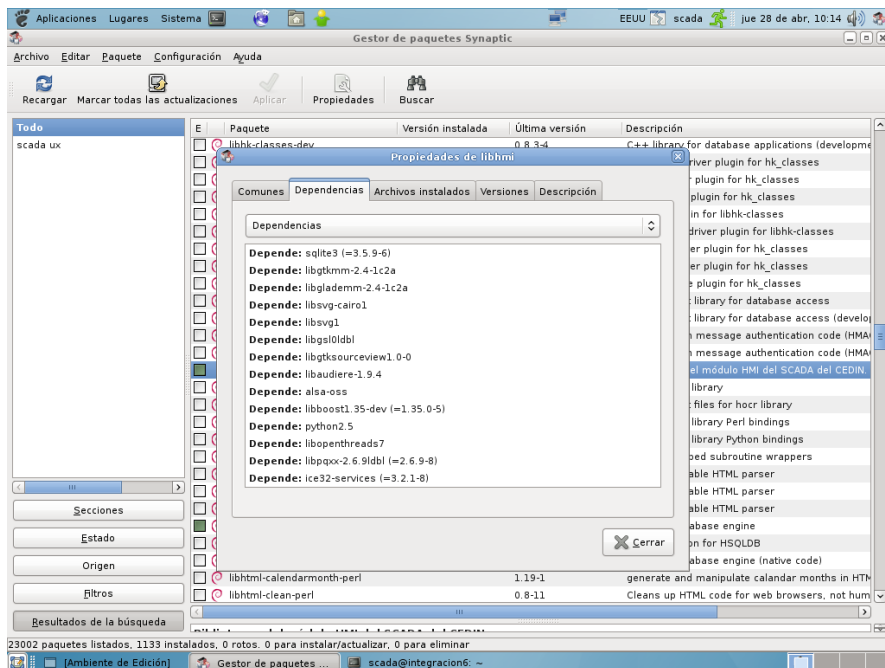
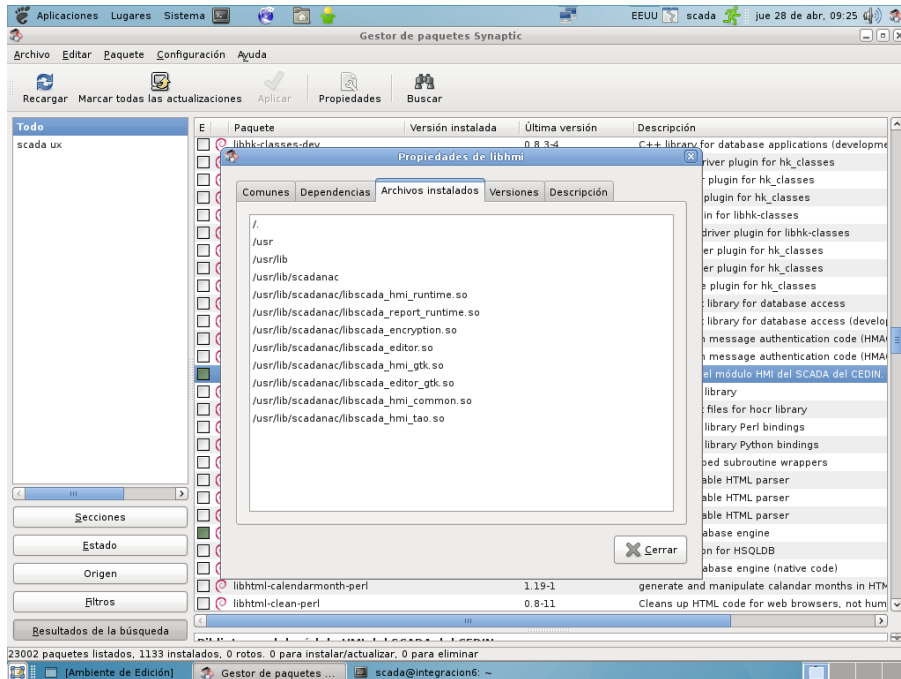
Anexo2: Métricas Externas propuestas

Métrica de Diagnosticabilidad	
Nombre de la métrica	Capacidad de análisis de fallos.
La métrica se propone medir	¿Puede el mantenedor fácilmente encontrar la causa del fallo?
Método de aplicación	Observe el comportamiento del operador que trata de resolver los fallos utilizando funciones de diagnóstico.
Medición (fórmula)	$X = 1 - A / B$ <p>A - número de fallos cuyas causas aún no han sido encontradas.</p> <p>B - número total de fallos registrados</p>
Interpretación del valor obtenido	$0 \leq X \leq 1$ Mientras más cercano al 1, mejor

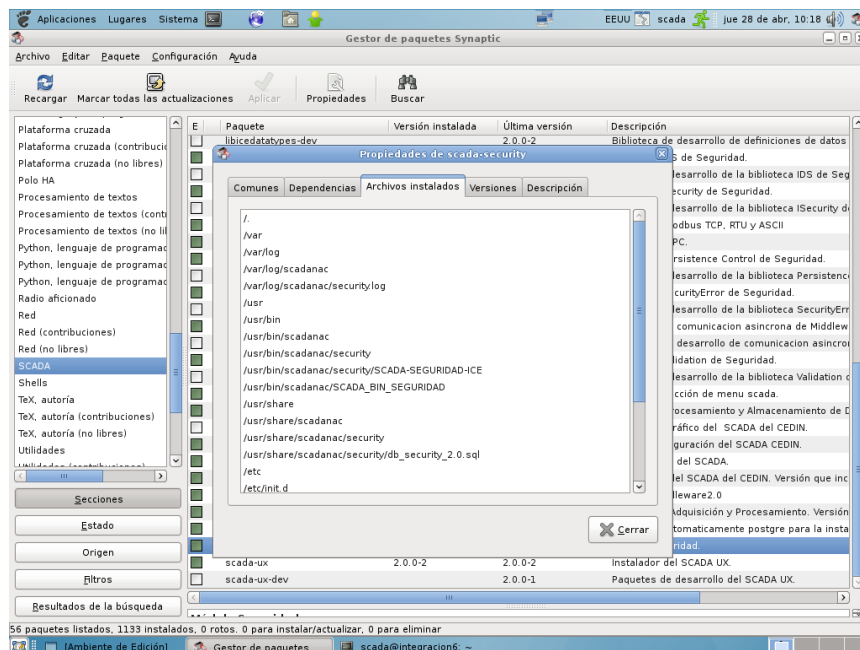
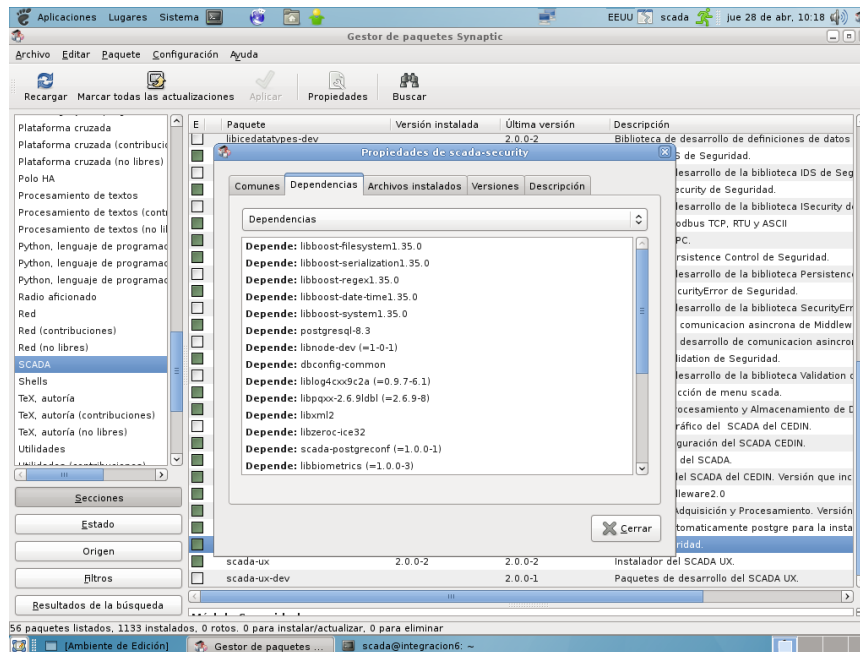
Métrica de Variabilidad	
Nombre de la métrica	Modificación Parametrizada
La métrica se propone medir	¿Puede el usuario o mantenedor cambiar el software a través de cambios de parámetros y resolver los problemas?
Método de aplicación	Observar el comportamiento del usuario o el responsable al tratar de cambiar el software.
Medición (fórmula)	$x = 1 - A / B$ <p>a = número de casos que el mantenedor intento cambiar el software mediante el parámetro.</p> <p>b = número de intentos para cambiar el software mediante el parámetro.</p>
Interpretación del valor obtenido	El más cercano a 1 es el mejor

Métrica de Cumplimiento de la mantenibilidad	
Nombre de la métrica	Cumplimiento de la mantenibilidad
La métrica se propone medir	¿Cuán conforme es la mantenibilidad del producto de software con los requisitos?
Método de aplicación	Cuenta el número de elementos que se encontraron en conformidad y compárelo con el número de elementos que requieren estar en conformidad según la especificación de requisitos.
Medición (fórmula)	$X = 1 - A / B$ A- número de elementos especificados que requiriendo estar en conformidad no han sido implementados. B - Número total de elementos especificados que requieren estar en conformidad.
Interpretación del valor obtenido	$0 \leq X \leq 1$ A mayor cercanía al 1 mejor

Anexo3: Instalación de las dependencias y archivos instalados del componente HMI



Anexo 4: Instalación de las dependencias y archivos instalados del componente Seguridad



Anexo 5: Elaboración del Informe de Evaluación de la Mantenibilidad de Componentes Software

Control de versiones

Fecha	Versión	Descripción	Autor
<dd/mm/aaaa >	<x.x>	<Detalles>	<Nombre y apellidos>

Introducción:

Descripción del componente < Nombre y breve descripción del componente a evaluar >

Estado del componente < Fase de desarrollo en que se encuentre el componente a evaluar >

Propósito < Objetivo de la evaluación >

Estructura organizativa < Roles involucrados >

Ambiente < Medios necesarios para realizar la evaluación >

Estándares < Estándares o guías usados para la evaluación >

Estrategia de evaluación < Especificación de la estrategia para evaluar la mantenibilidad según la fase en que se encuentre el componente, se aplican las métricas propuestas y los atributos a medir >

Nivel de aceptación de la evaluación < Establecimiento de niveles para clasificar los resultados en satisfactorio o insatisfactorio >

Categoría de los resultados < Categoría obtenida en los resultados de la evaluación de los atributos de la mantenibilidad >

Análisis de los resultados < Análisis de los resultados y elaboración de acciones correctivas ante las no conformidades encontradas >