

APRUEBAN DOCUMENTO “GUÍA TÉCNICA SOBRE EVALUACIÓN DE SOFTWARE PARA LA ADMINISTRACIÓN PÚBLICA”

RESOLUCIÓN MINISTERIAL N° 139-2004-PCM

Lima, 27 de mayo de 2004

CONSIDERANDO:

Que, mediante el Decreto Supremo N° 066-2003-PCM se fusionó la Subjefatura de Informática del Instituto Nacional de Estadística e Informática - INEI y la Presidencia del Consejo de Ministros, en virtud a lo cual el numeral 3.10 del artículo 3° del Reglamento de Organización y Funciones de la Presidencia del Consejo de Ministros, aprobado por Decreto Supremo N° 067-2003-PCM, ha establecido que es función de la Presidencia del Consejo de Ministros actuar como ente rector del Sistema Nacional de Informática;

Que, a efectos de implementar la infraestructura de Gobierno Electrónico, el mismo que comienza con la identificación y evaluación de los componentes funcionales requeridos, adopción de estándares abiertos y aceptados internacionalmente, la planificación y seguridad, en el marco de sus funciones la Oficina Nacional de Gobierno Electrónico e Informática – ONGEI, en coordinación con el Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual – INDECOPI, ha propuesto la “Guía Técnica Sobre Evaluación de Software para la Administración Pública”, por ser éste el que procesa datos y produce información, que es considerada actualmente un activo importante y estratégico de las organizaciones y países;

De conformidad con lo dispuesto por el Decreto Legislativo N° 560 – Ley del Poder Ejecutivo y el Reglamento de Organización y Funciones de la Presidencia del Consejo de Ministros, aprobado por Decreto Supremo N° 067-2003-PCM;

SE RESUELVE:

Artículo 1º.- Aprobar el documento “Guía Técnica Sobre Evaluación de Software para la Administración Pública”, documento que será publicado en el portal de la Presidencia del Consejo de Ministros (www.pcm.gob.pe).

Artículo 2º.- Las entidades de la Administración Pública, integrantes del Sistema Nacional de Informática, deberán aplicar lo establecido en la “Guía Técnica Sobre Evaluación de Software para la Administración Pública” en los productos de software que desarrollen o adquieran a partir de la fecha de publicación de la presente Resolución.

Si no fuera posible parcial o totalmente su aplicación, el área de informática o la que haga sus veces de la entidad respectiva, comunicará esta situación a la Oficina Nacional de Gobierno Electrónico e Informática – ONGEI, adjuntando el informe técnico que sustente la justificación para la no aplicación de la citada Guía.

Regístrese, comuníquese y publíquese.

CARLOS FERRERO
Presidente del Consejo de Ministros



**Oficina Nacional de Gobierno Electrónico e Informática
Presidencia del Consejo de Ministros**

**Guía Técnica sobre Evaluación
de Software en la
Administración Pública**

Presidencia del Consejo de Ministros – Gobierno del Perú – ONGEI
formatos@pcm.gob.pe

Ref: Guía Técnica de Evaluación
Versión: 01

Nombre del Proyecto: *“Guía Técnica sobre Evaluación de Software
en la Administración Pública”*



HOJA DE INFORMACION GENERAL

CONTROL DOCUMENTAL:

PROYECTO: Guía Técnica sobre Evaluación de Software en la Administración Pública
ENTIDAD: Presidencia del Consejo de Ministros
VERSIÓN: 1.0
FECHA EDICIÓN: 05/05/2004
NOMBRE DE ARCHIVO: P01-PCM-GUIAEVALUACIONSOFTWARE
RESUMEN: Guía que permite evaluar eficientemente los desarrollos y software en el Estado.

DERECHOS DE USO:

La presente documentación es de uso para la Administración Pública del Estado Peruano.

ESTADO FORMAL:

| | | |
|----------------------|----------------------|------------------|
| Preparado por: ONGEI | Entidad: ONGEI – PCM | Fecha: Mayo 2004 |
|----------------------|----------------------|------------------|



CONTROL DE VERSIONES

| FUENTE DE CAMBIO | FECHA DE SOLICITUD DEL CAMBIO | VERSIÓN | PARTES QUE CAMBIAN | DESCRIPCIÓN DEL CAMBIO | FECHA DE CAMBIO |
|------------------------------------|-------------------------------|---------|--------------------|------------------------|-----------------|
| P01-PCM-GUIAEVALUACIONSOFTWARE.doc | | 1.00 | N/A | | |



Índice

| | |
|---|-----------|
| INTRODUCCIÓN | 04 |
| APLICACIÓN | 04 |
| ESTRUCTURA | 04 |
| | |
| PARTE 1: MODELO DE LA CALIDAD | 05 |
| 1.1 Alcance | 05 |
| 1.2 Conformidad | 06 |
| 1.3 Marco de trabajo del modelo de la calidad | 06 |
| 1.4 Modelo de calidad para la calidad externa e interna | 11 |
| 1.5 Modelo de calidad para la calidad en uso | 18 |
| | |
| PARTE 2: MÉTRICAS | 20 |
| 2.1 Atributos Internos y Externos | 20 |
| 2.2 Métrica interna | 21 |
| 2.3 Métrica externa | 21 |
| 2.4 Relación entre las métricas internas y externas | 21 |
| 2.5 Calidad en el uso de métricas | 22 |
| 2.6 Opción de métrica y criterio de medidas | 22 |
| 2.7 Métricas usadas para la comparación | 23 |
| | |
| PARTE 3: PROCESO DE EVALUACIÓN DE SOFTWARE | 24 |
| 3.1 Establecer el propósito de la evaluación | 24 |
| 3.2 Identificar el tipo de producto | 24 |
| 3.3 Especificar el Modelo de Calidad | 24 |
| 3.4 Seleccionar métricas | 24 |
| 3.5 Establecer niveles, escalas para las métricas | 25 |
| 3.6 Establecer criterios de valoración | 25 |
| 3.7 Tomar medidas | 25 |
| 3.8 Comparar con los criterios | 26 |
| 3.9 Valorar resultados | 26 |
| 3.10 Documentación | 26 |
| | |
| GLOSARIO DE TÉRMINOS | 27 |
| | |
| BIBLIOGRAFÍA | 32 |



GUÍA TÉCNICA SOBRE EVALUACIÓN DE SOFTWARE PARA LA ADMINISTRACIÓN PÚBLICA

INTRODUCCIÓN

La presente guía esta basada sobre la norma ISO/IEC 9126 de la ISO (Organización Internacional de Normalización) y la IEC (Comisión Electrotécnica Internacional) que forman el sistema especializado para la normalización internacional.

El desarrollo o selección de productos de software con calidad es muy importante en la actualidad en las instituciones públicas, ya que éstas procesan información, que es considerada como un activo importante de sus organizaciones.

Una especificación y evaluación integral y detallada de la calidad de los productos de software es un factor clave para asegurar que la calidad sea la adecuada. Esto se puede lograr definiendo de manera apropiada las características de calidad, teniendo en cuenta el propósito del uso del producto de software en la institución.

Es importante especificar y evaluar cada característica relevante de la calidad de los productos de software, cuando esto sea posible, utilizando mediciones validadas o de amplia aceptación, que hagan técnicamente transparente esta actividad.

Agradecemos la colaboración del Comité Técnico de Normalización de Ingeniería de Software y Sistemas de Información - INDECOPI, por su apoyo técnico en la elaboración de la presente guía.

APLICACIÓN

- La presente guía es aplicable al **software propietario y software libre o de código abierto** utilizado en la Administración Pública.
- Esta guía debe aplicarse en toda evaluación de software propietario considerando esquemas comparativos con el software libre o de código abierto y viceversa, evidenciando ventajas y desventajas.
- Será utilizada para evaluar un solo software o un conjunto de softwares de naturaleza o funciones similares, tipo y/o categoría.

ESTRUCTURA

La presente guía consta de las siguientes partes:

- 1: Modelo de la calidad
- 2: Métricas



– 3. Proceso de evaluación de software

PARTE 1: MODELO DE LA CALIDAD

1.1 ALCANCE

Se describe un modelo de calidad para los productos de software, dividido en dos partes:

- a) Calidad interna y externa, y
- b) Calidad en uso.

La primera parte del modelo especifica seis características para calidad interna y externa, las cuales, a su vez, están subdivididas en sub características. Estas sub características se manifiestan externamente cuando el software es usado como parte de un sistema de computadora, y son el resultado de atributos internos de software.

La segunda parte del modelo, especifica cuatro características para la calidad en uso. Calidad en uso es el efecto combinado para el usuario de las seis características de la calidad interna y externa de productos de software.

Las características definidas son aplicables a todo software, incluyendo programas de computadora y datos contenidos en *firmware*. Las características y sub características proveen terminología consistente para la calidad de productos de software. Ellas también proveen un marco de trabajo para especificar los requerimientos de la calidad para productos de software, y para hacer análisis y evaluaciones entre capacidades de productos de software.

Esta parte de la norma permite especificar y evaluar la calidad de productos de software desde diferentes perspectivas asociadas con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría de software.

Esta norma será usada por personal de informática que cumple funciones de desarrolladores, adquirientes, personal de aseguramiento de la calidad y aquellos responsables de especificar y evaluar la calidad de productos de software.

Como ejemplo del uso del modelo de la calidad, tenemos:

- Validar que la definición de un requerimiento esté completa;
- Identificar requerimientos de software;
- Identificar objetivos del diseño de software;



- Identificar objetivos de prueba de software;
- Identificar criterios de aseguramiento de la calidad;
- Identificar criterios de aceptación para un producto de software completo.

1.2 CONFORMIDAD

Cualquier requerimiento, especificación o evaluación de la calidad sobre cualquier producto de software que cumpla esta parte de la guía, debe usar las características y sub características de los ítems 1.4 y 1.5, dando las razones por cualquier exclusión, o describiendo su propia categorización de los atributos de la calidad de productos de software, explicando la equivalencia respectiva.

1.3 MARCO DE TRABAJO DEL MODELO DE LA CALIDAD

En esta parte se describe el marco de trabajo de un modelo de calidad, el cual explica la relación entre los diferentes enfoques de la calidad.

1.3.1 Perspectivas de calidad

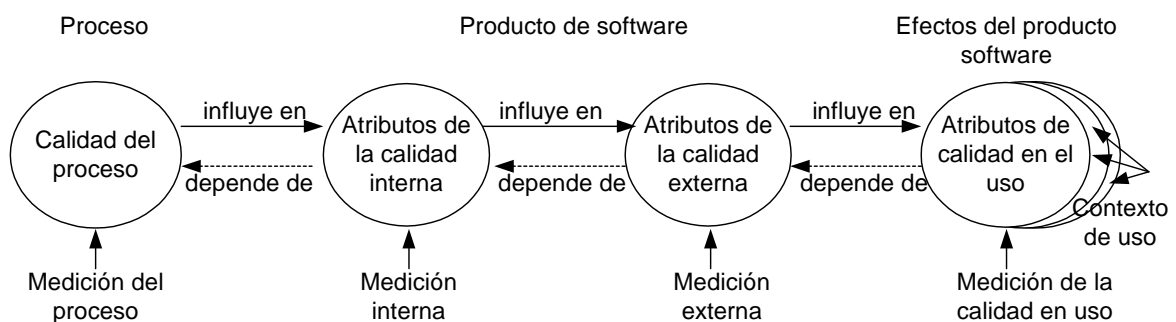


FIGURA 1 – Ciclo de vida de la calidad

Las necesidades de calidad del usuario incluyen requerimientos de calidad en uso, en contextos específicos. Estas necesidades identificadas pueden ser usadas cuando se especifiquen la calidad externa e interna, utilizando características y sub características de la calidad del producto de software.

La evaluación de los productos de software para satisfacer las necesidades de calidad es uno de los procesos en el ciclo de vida del desarrollo del software. La calidad del



producto de software puede ser evaluada midiendo atributos internos (medidas típicamente estáticas de productos intermedios), o midiendo atributos externos (midiendo típicamente el comportamiento del código cuando es ejecutado), o bien midiendo los atributos de aplicación de calidad en uso. El objetivo para que este producto tenga el efecto requerido en un contexto particular de uso se diagrama en la Figura 2.

La calidad del proceso contribuye a mejorar la calidad del producto, y la calidad del producto contribuye a mejorar la calidad en uso. Por lo tanto, evaluar y mejorar un proceso es una manera de mejorar la calidad del producto, y evaluar y mejorar la calidad del producto es una manera de mejorar la calidad en uso. De igual manera, evaluar la calidad en uso proporciona una retroalimentación para mejorar el producto, y evaluar un producto puede proporcionar una respuesta para mejorar un proceso.

Atributos internos apropiados en el software son pre requisitos para alcanzar el comportamiento externo requerido, y un apropiado comportamiento externo es un pre requisito para alcanzar la calidad en uso (Figura 1).

Los requisitos para la calidad del producto de software incluirán criterios de evaluación para calidad interna, calidad externa y calidad en uso, para cumplir las necesidades de los desarrolladores, responsables de mantenimiento, adquirientes y usuarios finales.

1.3.2 Calidad de producto y el ciclo de vida

Las vistas de calidad interna, calidad externa y calidad en uso cambian durante el ciclo de vida del software. Por ejemplo, la calidad especificada, como requisito de calidad al comienzo de un ciclo de vida, es mayormente observada desde el punto de vista externo y de usuario, y se diferencia de la calidad del producto intermedio, como la calidad del diseño, la cual es mayormente observada desde el punto de vista interno del desarrollador. Las tecnologías usadas para alcanzar el nivel de calidad necesario, así como la especificación y evaluación de calidad, necesitan soportar estos diversos puntos de vista. Es necesario definir estas perspectivas y las tecnologías asociadas a la calidad, para manejarla apropiadamente en cada etapa del ciclo de vida.

La meta es alcanzar la calidad necesaria y suficiente para cumplir con las necesidades reales de los usuarios. La norma ISO 8402 define calidad en términos de la habilidad de satisfacer necesidades explícitas (declaradas/descritas/especificadas) e implícitas. Sin embargo, las necesidades descritas por un usuario no siempre reflejan las verdaderas necesidades del mismo, porque:

- Un usuario normalmente no está consciente de sus necesidades reales.
- Las necesidades podrían cambiar después de ser especificadas.
- Diferentes usuarios pueden tener diferentes ambientes de operación.
- Podría ser imposible consultar a todos los posibles tipos de usuario, particularmente para un tipo de software (que no está en el mostrador/producto preelaborado).



Por lo tanto, los requisitos de calidad no pueden ser completamente definidos antes de empezar con el diseño. Sin embargo, es necesario entender las necesidades reales del usuario tan al detalle como sea posible, y representarlas en los requerimientos. La meta no es obtener la calidad perfecta, pero sí la calidad necesaria y suficiente para cada contexto específico de uso, cuando el producto sea entregado y utilizado por los usuarios.

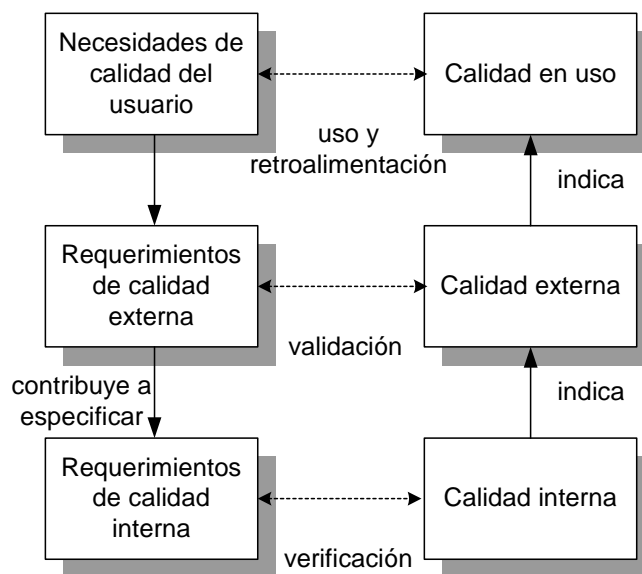


FIGURA 2 – Calidad en el ciclo de vida del software

Las escalas de medidas para las métricas usadas en los requerimientos de calidad pueden ser divididas en categorías correspondientes a diferentes grados de satisfacción de los requerimientos. Por ejemplo, la escala podría estar dividida en dos categorías: no satisfactoria y satisfactoria, o en cuatro categorías: excede los requerimientos, cumple los objetivos, mínimamente aceptable e inaceptable. Las categorías deberían ser especificadas para que ambos, el usuario y el desarrollador, puedan evitar costos innecesarios e incumplimiento de cronogramas. Existen diferentes perspectivas de la calidad del producto y sus métricas asociadas a las diferentes etapas del ciclo de vida del software. (Ver Figura 3)

Las **Necesidades de Calidad del Usuario** pueden ser especificadas como requerimientos de calidad por las métricas de calidad en uso, por métricas externas y a veces por métricas internas. Estos requerimientos especificados por las métricas, deberían ser usados como criterios cuando un producto es validado. Lograr un producto que satisfaga las necesidades del usuario, normalmente requiere de un enfoque interactivo en el desarrollo de software, con una continua retroalimentación desde la perspectiva del usuario.

Los **Requerimientos de Calidad Externos** especifican el nivel de calidad requerido desde una perspectiva externa. Estos incluyen requerimientos derivados de las necesidades de calidad de usuarios, incluyendo calidad en requerimientos de uso. Los



requerimientos de calidad externos son usados como los objetivos para la validación en varias etapas de desarrollo. Los requerimientos de calidad externos para todas las características de calidad definidas en esta parte, deben ser establecidos en la especificación de requerimientos de calidad usando métricas externas, deben ser transformados en requerimientos de calidad internos y deben ser usados como criterios cuando un producto es evaluado.

Los **Requerimientos de Calidad Internos** especifican el nivel de calidad requerido desde la perspectiva interna del producto. Los requerimientos de calidad internos son usados para especificar propiedades internas de productos. Estos pueden incluir modelos estáticos y dinámicos, otros documentos y código fuente. Los requerimientos de calidad internos pueden ser usados como objetivos para la validación en varias etapas de desarrollo. Ellos también pueden ser usados para definir estrategias de desarrollo y criterios de evaluación y verificación durante el desarrollo. Esto puede incluir el uso de métricas adicionales (por ejemplo: reusabilidad). Los requerimientos específicos de calidad interna deben ser especificados cuantitativamente usando métricas internas.

La **Calidad Interna** es la totalidad de características del producto de software desde una perspectiva interna. La calidad interna es medida y evaluada en base a los requerimientos internos de calidad. Los detalles de la calidad del producto de software pueden ser mejorados durante la implementación, revisión y prueba del código fuente del software, pero la naturaleza fundamental de la calidad del producto de software representada por la calidad interna, permanece sin cambios a menos que sea rediseñado.

La **Calidad Externa Estimada (o Predicha)** es la calidad que es estimada o predicha para el producto de software final, en cada etapa de desarrollo para cada característica de calidad, basada en el conocimiento de la calidad interna.

La **Calidad Externa** es la totalidad de las características del producto de software desde una perspectiva externa. Es la calidad cuando el software es ejecutado, la cual es típicamente medida y evaluada en un ambiente simulado, con datos simulados y usando métricas externas. Durante las pruebas, muchas fallas serán descubiertas y eliminadas. Sin embargo, algunas fallas todavía pueden permanecer después de las pruebas. Como es difícil corregir la arquitectura del software u otros aspectos fundamentales del diseño del software, el diseño fundamental permanece sin cambios a través de las pruebas.

La **Calidad en Uso Estimada (o Predicha)** es la calidad que es estimada o predicha para el producto de software final, en cada etapa de desarrollo para cada característica de calidad en uso, y se basa en el conocimiento de la calidad externa e interna.

La calidad externa y la calidad en uso pueden ser estimadas y predichas durante el desarrollo de cada característica de calidad cuando las tecnologías apropiadas son desarrolladas. Sin embargo, como actualmente no se proporciona todo el soporte necesario para el propósito de predicción, se debe desarrollar más tecnología para mostrar la correlación entre la calidad interna, la calidad externa y la calidad en uso.



La **Calidad en Uso** es la perspectiva del usuario de la calidad del producto de software cuando éste es usado en un ambiente específico y en un contexto de uso específico. Esta mide la extensión en la cual los usuarios pueden conseguir sus metas en un ambiente particular, en vez de medir las propiedades del software en si mismo.

El término 'Usuario' se refiere a cualquier tipo de posible usuario, incluyendo operadores y personal de mantenimiento, y sus requerimientos pueden ser diferentes.

El nivel de calidad en el ambiente del usuario puede ser diferente del ambiente de desarrollo, debido a diferencias entre las necesidades y capacidades de diversos usuarios y diferencias entre hardware y ambientes de soporte. El usuario evalúa sólo aquellos atributos de software que son usados para sus tareas. Algunas veces, los atributos de software especificados por un usuario final durante la fase de análisis de requerimientos, ya no cumplen los requerimientos del usuario cuando el producto está en uso, debido a cambiantes requerimientos del usuario y a la dificultad de especificar necesidades implícitas.

1.3.3 Ítems a ser evaluados

Los ítems pueden ser evaluados por medición directa, o de manera indirecta, midiendo sus consecuencias. Por ejemplo, un proceso puede ser medido indirectamente por la medición y evaluación de sus productos, y un producto puede ser evaluado indirectamente por la medición del desempeño de un usuario en sus tareas (usando métricas de calidad en uso).

El software nunca corre solo sino que siempre es parte de un sistema mayor, típicamente consistente de otros productos de software con los cuales él tiene interfaces: hardware, operadores humanos, y flujos de trabajo. El producto de software completado puede ser evaluado por los niveles de las métricas externas elegidas. Estas métricas describen su interacción con su entorno, y son medidas al observar el software en operación. La calidad en uso puede ser medida por la extensión por la cual un producto empleado por usuarios específicos cumple las necesidades de alcanzar metas específicas con efectividad, productividad, seguridad y satisfacción. Esto normalmente será complementado con mediciones de características de calidad más específicas del producto de software, lo cual también es posible en el proceso inicial de desarrollo.

En etapas más tempranas de desarrollo, sólo pueden ser medidos los recursos y procesos. Cuando los productos intermedios (especificaciones, código fuente, etc.) se tornan disponibles, estos pueden ser evaluados por los niveles de las métricas internas elegidas. Estas métricas pueden ser usadas para predecir los valores de las métricas externas. Ellas también pueden ser medidas por derecho propio, al ser pre requisitos esenciales para la calidad externa.

Se puede hacer una distinción adicional entre la evaluación del producto de software y la evaluación del sistema en el cual es ejecutado.

Por ejemplo, la confiabilidad de un sistema es medida al observar todas las fallas originadas por cualquier causa (hardware, software, errores humanos, etc.), mientras que la confiabilidad del producto de software es medida al extraer de las fallas



observadas sólo aquellas que son debidas a faltas en el software (originadas en requerimientos, diseño o implementación).

Además, depende del propósito de la evaluación y de quienes son los usuarios, el juzgar dónde están los límites del sistema.

En ese sentido, por ejemplo, si se supone que los pasajeros son los usuarios de un avión con un sistema de control de vuelo basado en computadora, entonces el sistema del cual ellos dependen incluye la tripulación, el fuselaje, el hardware y software del sistema de control de vuelo, mientras que si se toma a la tripulación como los usuarios, entonces el sistema del cual ellos dependen consiste sólo del fuselaje y el sistema de control de vuelo.

1.3.4 Usando un modelo de calidad

La calidad de un producto de software se debe evaluar usando un modelo definido. El modelo de calidad debe ser utilizado al fijar las metas de la calidad para los productos de software y los productos intermedios. La calidad del producto de software debería ser jerárquicamente descompuesta en un modelo de calidad constituido por características y sub características, las cuales se pueden utilizar como lista de comprobación de las ediciones relacionadas con la calidad. Más adelante se define un modelo jerárquico de calidad (aunque otras maneras de categorizar la calidad pueden ser más apropiadas en circunstancias particulares y justificadas).

No es prácticamente posible medir todas las sub características internas y externas para todas las partes de un gran producto de software. De modo similar, no es generalmente práctico medir la calidad en el uso para todos los escenarios posibles de las tareas del usuario. Los recursos para la evaluación necesitan ser asignados entre los diversos tipos de medida, dependiente de los objetivos de la institución y de la naturaleza del producto y diseño de procesos.

1.4 Modelo de calidad para la calidad externa e interna

En esta sección se define el Modelo de Calidad para la calidad externa e interna a ser usado en las instituciones públicas. Se han establecido categorías para las cualidades de la calidad del software, basadas en seis características (funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento y portabilidad), que se subdividen a su vez en sub características (Figura 3). Las sub características se pueden medir por métrica interna o externa.

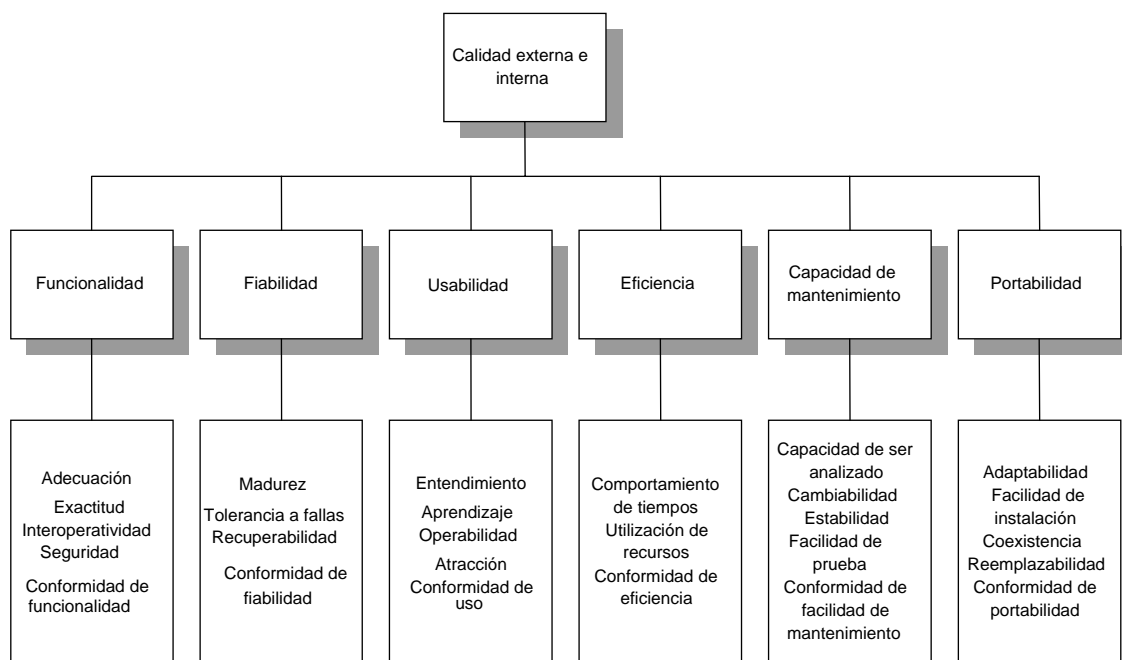


FIGURA 3 – Modelo de calidad para la calidad externa e interna

Las definiciones se dan para cada característica y sub característica de calidad del software que influye en la calidad. Para cada característica y sub característica, la capacidad del software es determinada por un conjunto de atributos internos que pueden ser medidos. Las características y sub características se pueden medir externamente por la capacidad provista por el sistema que contiene el software.

1.4.1 Funcionalidad

La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas.

Esta característica se refiere a lo que hace el software para satisfacer necesidades, mientras que las otras características se refieren principalmente a cuándo y a cómo satisfacen las necesidades.

Para un sistema que es operado por un usuario, la combinación de la funcionalidad, fiabilidad, usabilidad y eficiencia puede ser medida externamente por su calidad en uso.

1.4.1.1 Adecuación

La capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario.

Ejemplos de adecuación son la composición orientada a tareas de funciones a partir de sub funciones que las constituyen, y las capacidades de las tablas.



1.4.1.2 Exactitud

La capacidad del producto de software para proveer los resultados o efectos acordados con un grado necesario de precisión.

1.4.1.3 Interoperabilidad

La capacidad del producto de software de interactuar con uno o más sistemas especificados. La interoperabilidad se utiliza en lugar de compatibilidad para evitar una posible ambigüedad con la reemplazabilidad.

1.4.1.4 Seguridad

La capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos, y a las personas o sistemas autorizados no se les niegue el acceso a ellos.

La seguridad en un sentido amplio se define como característica de la calidad en uso, pues no se relaciona con el software solamente, sino con todo un sistema.

1.4.1.5 Conformidad de la funcionalidad

La capacidad del producto de software de adherirse a los estándares, convenciones o regulaciones legales y prescripciones similares referentes a la funcionalidad.

1.4.2 Fiabilidad

La capacidad del producto de software para mantener un nivel específico de funcionamiento cuando se está utilizando bajo condiciones especificadas.

El desgaste o envejecimiento no ocurre en el software. Las limitaciones en fiabilidad son debido a fallas en los requerimientos, diseño, e implementación. Las fallas debido a estos errores dependen de la manera en que se utiliza el producto de software y de las opciones del programa seleccionadas, más que del tiempo transcurrido.

La definición de fiabilidad en la ISO/IEC 2382-14:1997 es "la habilidad de la unidad funcional de realizar una función requerida...". En este documento, la funcionalidad es solamente una de las características de la calidad del software. Por lo tanto, la definición de la fiabilidad se ha ampliado a "mantener un nivel especificado del funcionamiento..." en vez de "...realizar una función requerida".

1.4.2.1 Madurez

La capacidad del producto de software para evitar fallas como resultado de errores en el software.



1.4.2.2 Tolerancia a errores

La capacidad del producto de software para mantener un nivel especificado de funcionamiento en caso de errores del software o de incumplimiento de su interfaz especificada.

El nivel especificado de funcionamiento puede incluir la falta de capacidad de seguridad.

1.4.2.3 Recuperabilidad

La capacidad del producto de software para restablecer un nivel especificado de funcionamiento y recuperar los datos afectados directamente en el caso de una falla.

Después de una falla, un producto de software a veces estará no disponible por cierto período del tiempo, intervalo en el cual se evaluará su recuperabilidad.

La disponibilidad es la capacidad del producto de software para poder realizar una función requerida en un punto dado en el tiempo, bajo condiciones indicadas de uso. En extremo, la disponibilidad se puede determinar por la proporción de tiempo total, durante la cual, el producto de software está en un estado ascendente. La disponibilidad, por lo tanto, es una combinación de madurez (con control de frecuencias de fallas), de la tolerancia de errores y de la recuperabilidad (que gobierna el intervalo de tiempo en cada falla). Por esta razón es que no ha sido incluida como una sub característica separada.

1.4.2.4 Conformidad de la fiabilidad

La capacidad del producto de software para adherirse a las normas, convenciones o regulaciones relativas a la fiabilidad.

1.4.3 Usabilidad

La capacidad del producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo las condiciones especificadas.

Algunos aspectos de funcionalidad, fiabilidad y eficiencia también afectarán la usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no son clasificados como usabilidad.

Los usuarios pueden ser operadores, usuarios finales y usuarios indirectos que están bajo la influencia o dependencia del uso del software. La usabilidad debe dirigirse a todo los diferentes ambientes de usuarios que el software puede afectar, o estar relacionado con la preparación del uso y evaluación de los resultados.



1.4.3.1 Entendimiento

La capacidad del producto de software para permitir al usuario entender si el software es adecuado, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación.

Esto dependerá de la documentación y de las impresiones iniciales dadas por el software.

1.4.3.2 Aprendizaje

La capacidad del producto de software para permitir al usuario aprender su aplicación. Un aspecto importante a considerar aquí es la documentación del software.

1.4.3.3 Operabilidad

La capacidad del producto de software para permitir al usuario operarlo y controlarlo.

Los aspectos de propiedad, de cambio, de adaptabilidad y de instalación pueden afectar la operabilidad.

La operabilidad corresponde a la controlabilidad, a la tolerancia a errores y a la conformidad con las expectativas del usuario.

Para un sistema que es operado por un usuario, la combinación de la funcionalidad, confiabilidad, usabilidad y eficacia puede ser una medida considerada por la calidad en uso.

1.4.3.4 Atracción

La capacidad del producto de software de ser atractivo al usuario.

Esto se refiere a las cualidades del software para hacer el software más atractivo al usuario, tal como el uso del color y la naturaleza del diseño gráfico.

1.4.3.5 Conformidad de uso

La capacidad del producto de software para adherirse a los estándares, convenciones, guías de estilo o regulaciones relacionadas a su usabilidad.

1.4.4 Eficiencia

La capacidad del producto de software para proveer un desempeño adecuado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas.

Los recursos pueden incluir otros productos de software, la configuración de hardware y software del sistema, y materiales (Ej: Papel de impresión o diskettes).



Para un sistema operado por usuarios, la combinación de funcionalidad, fiabilidad, usabilidad y eficiencia pueden ser medidas externamente por medio de la calidad en uso.

1.4.4.1 Comportamiento de tiempos

La capacidad del producto de software para proveer tiempos adecuados de respuesta y procesamiento, y ratios de rendimiento cuando realiza su función bajo las condiciones establecidas.

1.4.4.2 Utilización de recursos

La capacidad del producto de software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo las condiciones establecidas.

Los recursos humanos están incluidos dentro del concepto de productividad.

1.4.4.3 Conformidad de eficiencia

La capacidad del producto de software para adherirse a estándares o convenciones relacionados a la eficiencia.

1.4.5 Capacidad de mantenimiento

Capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y especificaciones de requerimientos funcionales.

1.4.5.1 Capacidad de ser analizado

La capacidad del producto de software para atenerse a diagnósticos de deficiencias o causas de fallas en el software o la identificación de las partes a ser modificadas.

1.4.5.2 Cambiabilidad

La capacidad del software para permitir que una determinada modificación sea implementada.

Implementación incluye codificación, diseño y documentación de cambios.

Si el software va a ser modificado por el usuario final, la cambiabilidad podría afectar la operabilidad.

1.4.5.3 Estabilidad

La capacidad del producto de software para evitar efectos inesperados debido a modificaciones del software.



1.4.5.4 Facilidad de prueba

La capacidad del software para permitir que las modificaciones sean validadas.

1.4.5.5 Conformidad de facilidad de mantenimiento

La capacidad del software para adherirse a estándares o convenciones relativas a la facilidad de mantenimiento.

1.4.6 Portabilidad

La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software.

1.4.6.1 Adaptabilidad

La capacidad del producto de software para ser adaptado a diferentes entornos especificados sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado.

Adaptabilidad incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).

Si el software va a ser adaptado por el usuario final, la adaptabilidad corresponde a la conveniencia de la individualización, y podría afectar la operabilidad.

1.4.6.2 Facilidad de instalación

La capacidad del producto de software para ser instalado en un ambiente especificado.

Si el software va a ser instalado por el usuario final, puede afectar la propiedad y operatividad resultantes.

1.4.6.3 Coexistencia

La capacidad del producto de software para coexistir con otros productos de software independientes dentro de un mismo entorno, compartiendo recursos comunes.

1.4.6.4 Reemplazabilidad

La capacidad del producto de software para ser utilizado en lugar de otro producto de software, para el mismo propósito y en el mismo entorno.

Por ejemplo, la reemplazabilidad de una nueva versión de un producto de software es importante para el usuario cuando dicho producto de software es actualizado (actualizaciones, upgrades).



Reemplazabilidad se utiliza en lugar de compatibilidad de manera que se evitan posibles ambigüedades con la interoperabilidad.

La reemplazabilidad puede incluir atributos de ambos, inestabilidad y adaptabilidad. El concepto ha sido introducido como una sub característica por sí misma, dada su importancia.

1.4.6.5 Conformidad de portabilidad

La capacidad del software para adherirse a estándares o convenciones relacionados a la portabilidad.

1.5 MODELO DE CALIDAD PARA LA CALIDAD EN USO

En esta parte se define el modelo de calidad para la calidad en uso. Los atributos de la calidad en uso están categorizados en cuatro características: eficacia, productividad, seguridad y satisfacción (Figura 4).

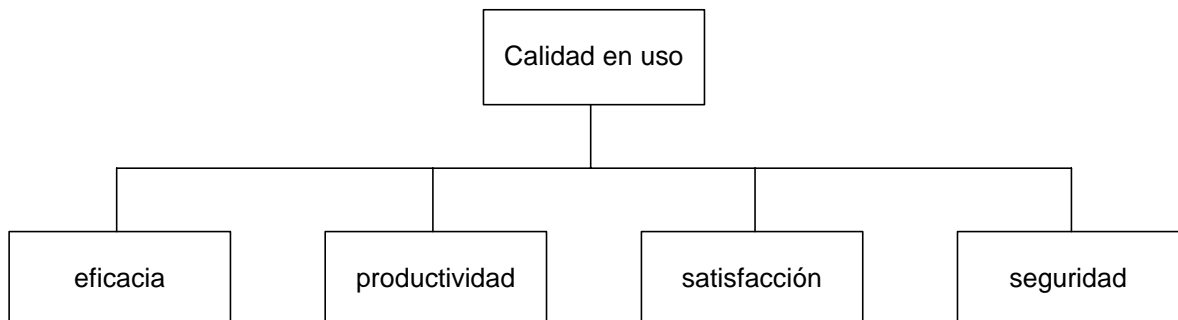


FIGURA 4 – Modelo de calidad para la calidad en uso

La calidad en uso es la visión de calidad del usuario. Alcanzar la calidad en uso depende de alcanzar la calidad externa necesaria que a su vez depende de alcanzar la calidad interna necesaria (Figura 1)

Las medidas son normalmente requeridas en tres niveles: interno, externo y de uso. Encontrar criterios para las medidas internas, no es normalmente suficiente para asegurar el logro de criterios para las medidas externas, y encontrar criterios para las medidas externas, no es normalmente suficiente para asegurar el logro de criterios para la calidad en uso.



1.5.1 Calidad en uso

La capacidad del producto de software para permitirles a usuarios específicos lograr las metas propuestas con eficacia, productividad, seguridad y satisfacción, en contextos especificados de uso.

Calidad en uso es la visión de calidad del usuario de un entorno que contiene el software, y es medida a partir de los resultados de usar el software en el entorno, más que por las propiedades del software mismo.

1.5.1.1 Eficacia

La capacidad del producto de software para permitir a los usuarios lograr las metas especificadas con exactitud e integridad, en un contexto especificado de uso.

1.5.1.2 Productividad

La capacidad del producto de software para permitir a los usuarios emplear cantidades apropiadas de recursos, en relación a la eficacia lograda en un contexto especificado de uso.

Los recursos relevantes pueden incluir: tiempo para completar la tarea, esfuerzo del usuario, materiales o costo financiero.

1.5.1.3 Seguridad

La capacidad del producto de software para lograr niveles aceptables de riesgo de daño a las personas, institución, software, propiedad (licencias, contratos de uso de software) o entorno, en un contexto especificado de uso.

Los riesgos son normalmente el resultado de deficiencias en la funcionalidad (incluyendo seguridad), fiabilidad, usabilidad o facilidad de mantenimiento.

1.5.1.4 Satisfacción

La capacidad del producto de software para satisfacer a los usuarios en un contexto especificado de uso.

La satisfacción es la respuesta del usuario a la interacción con el producto, e incluye las actitudes hacia el uso del producto.



PARTE 2: MÉTRICAS

2.1 Atributos Internos y Externos

Los niveles de ciertos atributos internos se han encontrado para influir en los niveles de algunos atributos externos, de modo que haya un aspecto externo y un aspecto interno en la mayoría de las características. Por ejemplo, la confiabilidad puede ser medida externamente observando el número de fallas en un período dado del tiempo de ejecución durante un ensayo del software, e internamente examinando las especificaciones detalladas y el código fuente para determinar el nivel de la tolerancia de falla. Los atributos internos serían los indicadores de los atributos externos.

Un atributo interno puede influenciar a una o más características, y una característica puede estar influenciada por más de un atributo (ver Figura 5). En este modelo la totalidad de atributos de la calidad del producto de software se clasifica en una estructura arborescente jerárquica de características y de sub características. El nivel más alto de esta estructura consiste en características de calidad y el nivel más bajo consiste en atributos de calidad de software. La jerarquía no es perfecta cuando algunos atributos pueden contribuir a más de una sub característica.

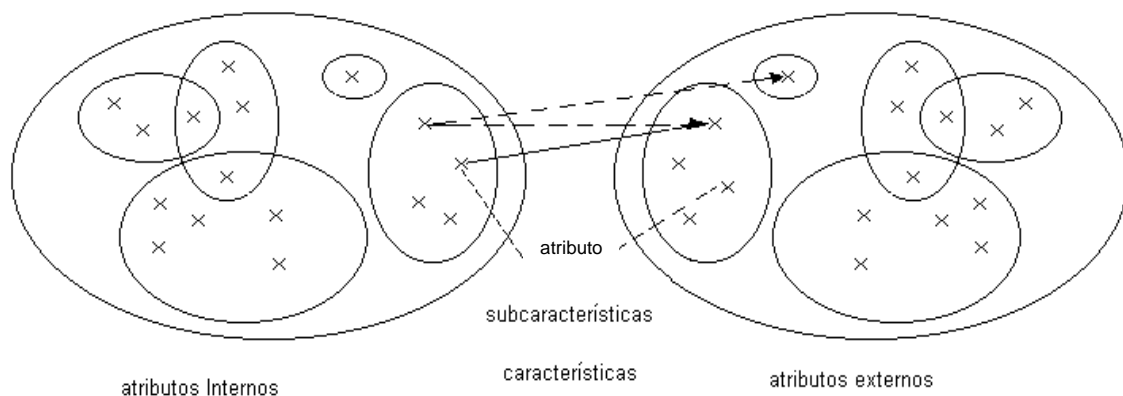


Figura 5 – Características de calidad, sub características y atributos

La sub característica puede medirse por la métrica interna o por la métrica externa.

La correlación entre los atributos internos y las medidas externas nunca es perfecta, y el efecto que un atributo interno dado tiene en una medida externa asociada, será determinado por la experiencia, y dependerá del contexto particular en que el software es usado.

De la misma manera, las propiedades externas (como la conveniencia, exactitud, tolerancia a fallas o tiempos de ejecución) influirán en la calidad observada. Una falla en la calidad del uso (por ejemplo: el usuario no puede completar la tarea) puede



remontarse a los atributos de calidad externa (por ejemplo: conveniencia u operabilidad) y los atributos internos asociados tienen que ser cambiados.

2.2 Métrica interna

La métrica interna puede ser aplicada a un producto de software no-ejecutable (como una especificación o código fuente) durante el diseño y la codificación. En el desarrollo de un producto de software, los productos intermedios deben ser evaluados usando métricas internas que permitan medir las propiedades intrínsecas, incluyendo aquellas que pueden derivarse de comportamientos simulados. El propósito primario de esta métrica interna es asegurar que se logre la calidad externa y la calidad de uso requerida. La métrica interna proporciona a los usuarios, evaluadores, verificadores y desarrolladores el beneficio de que puedan evaluar la calidad del producto de software y lo referido a problemas de calidad antes que el producto de software sea puesto en ejecución.

Las métricas internas miden atributos internos o indican los atributos externos, a través del análisis de las propiedades estáticas de productos intermedios o entregables del software. Las medidas de las métricas internas usan números o frecuencias de elementos de composición de software, los cuales aparecen, por ejemplo, en las sentencias de código de fuente, control de gráficos, flujo de datos y estados de representación de procesos.

2.3 Métrica externa

Las métricas externas usan medidas de un producto de software, derivadas del comportamiento del mismo, a través de la prueba, operación y observación del software. Antes de adquirir o usar un producto de software, éste debe ser evaluado usando las métricas basadas en los objetivos del área usuaria de la institución relacionados al uso, explotación y dirección del producto, considerando la organización y el ambiente técnico. La métrica externa proporciona a los usuarios, evaluadores, verificadores y desarrolladores, el beneficio de que puedan evaluar la calidad del producto de software durante las pruebas o el funcionamiento.

2.4 Relación entre las métricas internas y externas

Cuando los requisitos de calidad del producto de software son definidos, se listan las características o sub características de calidad del producto de software que contribuyen a dichos requisitos. Entonces, las métricas externas apropiadas y los rangos aceptables son especificados para cuantificar el criterio de calidad que valida que el software satisface las necesidades del usuario. Luego, los atributos de calidad interna del software se definen y especifican para planear y finalmente lograr la calidad externa y calidad en el uso requeridas, para construirlos durante el desarrollo del producto.

Apropiadas métricas internas y rangos aceptables son especificados para cuantificar los atributos de calidad interna, así ellos pueden usarse para verificar que el software intermedio reúne las especificaciones de calidad interna durante el desarrollo.



Se recomienda que las métricas internas que se usen tengan en lo posible una fuerte relación con la métrica externa diseñada, para que ellas puedan ser usadas para predecir los valores de las métricas externas. Sin embargo, es generalmente difícil diseñar un modelo teórico riguroso que proporcione una relación fuerte entre la métrica interna y la externa.

2.5 Calidad en el uso de métricas

La calidad en el uso de métricas mide la extensión de un producto que reúne las necesidades especificadas por los usuarios para lograr las metas propuestas, con la efectividad, productividad, seguridad y satisfacción en un contexto de uso específico. La evaluación de la calidad en uso valida la calidad del producto de software en los escenarios específicos de tareas de usuario.

La calidad en el uso es la vista del usuario sobre la calidad que el sistema de software contiene y es medida en términos de resultados de uso del software, en lugar de las propiedades del propio software. La calidad en el uso es el efecto combinado de calidad interna y externa para el usuario.

La relación de calidad en el uso con otras características de calidad del producto de software depende del tipo de usuario:

- El usuario final para quien la calidad en el uso es principalmente un resultado de funcionalidad, fiabilidad, utilidad y eficacia.
- La persona que mantiene el software para quien la calidad en el uso es un resultado del mantenimiento.
- La persona que hace portable el software para quien la calidad en el uso es un resultado de portabilidad.

2.6 Opción de métrica y criterio de medidas

La base en que las métricas son seleccionadas dependerá de las metas de la institución para el producto y las necesidades del evaluador. Las necesidades son especificadas por un criterio de medidas. El modelo en esta parte soporta una variedad de requisitos de evaluación, por ejemplo:

- Un usuario o un área de la institución, podría evaluar la conveniencia de un producto de software usando las métricas de calidad en el uso.
- Una institución que adquiere, podría evaluar un producto de software contra un criterio de valores de medidas externas de funcionalidad, fiabilidad, utilidad y eficacia, o de calidad en el uso.
- Un responsable de mantenimiento, podría evaluar un producto de software usando métricas para mantenimiento.
- Una persona responsable de la implementación del software en diferentes ambientes, podría evaluar un producto de software usando métricas de portabilidad.
- Un desarrollador, podría evaluar un producto de software contra criterios de



valores usando medidas internas de cualquiera de las características de calidad.

2.7 Métricas usadas para la comparación

Al informar los resultados del uso de métricas cuantitativas para hacer las comparaciones entre los productos, el informe mostrará si las métricas son objetivas o empíricas, usando valores conocidos y reproducibles.

Las comparaciones fiables entre los productos sólo se pueden hacer cuando se usan métricas rigurosas. Los procedimientos de medición deben medir las características (o sub características) de calidad del producto de software. Estos exigen ser medidos con suficiente exactitud para permitir asignar los criterios y hacer las comparaciones.

La concesión debe hacerse para posibles errores de medición causados por herramientas de medida o errores humanos.

La métrica usada para las comparaciones debe ser válida y suficientemente exacta para permitir hacer comparaciones fiables. Esto significa que las medidas deben ser objetivas, empíricas, usando una escala válida, y reproducibles.

- Para ser objetivo, habrá un procedimiento escrito y convenido para asignar el número o categoría al atributo del producto.
- Para ser empírico, los datos serán obtenidos de la observación o de un cuestionario psicométricamente válido.
- Para utilizar una escala válida, los datos deberán estar basados en ítems de igual valor o de un valor conocido. Si una lista de comprobación se utiliza para proporcionar datos, los ítems deben, si es necesario, ser ponderados.
- Para ser reproducible, el proceso para medir debería producir las mismas medidas (dentro de las tolerancias apropiadas) que son obtenidas por diferentes personas haciendo la misma medición del producto de software en diferentes ocasiones.

Las métricas internas deberían también tener valor predictivo, esto es, ellas deben correlacionarse con algunas medidas externas deseadas. Por ejemplo, una medida interna de un atributo particular del software debería tener correlación con cierto aspecto medible de calidad cuando se utiliza el software. Es importante que los valores asignados a las mediciones coincidan con las expectativas normales. Por ejemplo, si la medición sugiere que el producto es de alta calidad, entonces ésta debería ser consistente con el producto, satisfaciendo las necesidades de un usuario.



PARTE 3: PROCESO DE EVALUACIÓN DE SOFTWARE

Todo proceso de evaluación de software deberá partir de una evaluación cualitativa y derivar en una evaluación cuantitativa, siendo todo el proceso documentado, cumpliendo los siguientes pasos:

3.1 Establecer el propósito de la evaluación:

Productos intermedios:

- Decidir sobre la aceptación de un producto intermedio de un subcontratista o proveedor.
- Decidir cuándo un proceso está completo y cuando remitir los productos al siguiente proceso.
- Predecir o estimar la calidad del producto final.
- Recoger información con objeto de controlar y gestionar el proceso.
- Otros con justificación.

Producto final:

- Decidir sobre la aceptación del producto.
- Decidir cuando publicar el producto.
- Comparar el producto con otros productos competitivos.
- Seleccionar un producto entre productos alternativos.
- Valorar tanto el aspecto positivo, como el negativo, cuando está en uso.
- Decidir cuando mejorar o reemplazar un producto.
- Otros con justificación.

3.2 Identificar el tipo de producto

Especificar el tipo de producto a evaluar, si es un sistema operativo, software de seguridad, software de ofimática, lenguaje de programación, base de datos, aplicativo desarrollado, ERP, entre otros. Asimismo, se deberá establecer su relación con Estándares de Tecnologías de Información y Comunicaciones que utiliza la Institución; y asegurar la legalidad del producto.

3.3 Especificar el Modelo de Calidad

Se elaborará de acuerdo a lo establecido en la Parte I, y deberá ser aprobado por el Jefe de Informática o quien haga sus veces.

3.4 Seleccionar métricas

La selección de métricas se obtiene a partir de los atributos especificados en el Modelo de Calidad. Se agruparán en:



- Métricas internas.
- Métricas externas.
- Métricas de uso.

3.5 Establecer niveles, escalas para las métricas

- El área de informática aplicará el tipo de escala de proporción.
- A cada métrica seleccionada le asignará un puntaje máximo de referencia.
- La suma de los puntajes máximos de todas las métricas deberá ser igual a 100 puntos.
- El área de informática podrá establecer niveles de calificación cualitativa en base a los puntajes como por ejemplo:
 - Puntaje mínimo de aprobación.
 - Inaceptable, mínimo aceptable, rango objeto, excede los requisitos.
 - Insatisfactorio, satisfactorio.
- Se pueden usar números hasta con un decimal de aproximación. (Ejemplos: 4.1, 3.8, 11.7).
- El área de informática podrá establecer, por cada métrica, un puntaje mínimo de aprobación. En caso no se alcance ese puntaje, se considerará que el producto de software no cumple con las necesidades de información de la institución y será rechazado.

3.6 Establecer criterios de valoración

El área de informática elaborará sus procedimientos, con criterios distintos para diferentes características de calidad, cada uno puede estar expresado en términos de sub características individuales, o una combinación ponderada de ellas. El procedimiento puede incluir otros aspectos como el tiempo y costo que contribuyen a la estimación de la calidad de un producto de software en un entorno concreto.

3.7 Tomar medidas

Para la medición, las métricas seleccionadas se aplican al producto de software. Los resultados son valores expresados en las escalas de las métricas, definidos previamente.



3.8 Comparar con los criterios

En el paso de puntuación, el valor medido se compara con los criterios predeterminados.

Se debe elaborar un cuadro de resultados, como el que se aprecia a continuación.

| | PUNTAJE MAX. | SOFT. 1 | SOFT. 2 | | SOFT.n |
|--|--|----------------|----------------|--------------|---------------|
| Atributos internos (Ai) <ul style="list-style-type: none"> • Ai1 • Ai2 • . • . • Ain | PMax. Ai1 PMax. Ai2 . . PMax Ain | | | | |
| Atributos externos (Ae) <ul style="list-style-type: none"> • Ae1 • Ae2 • . • . • Aen | PMax Ae1 PMax Ae2 . . PMax Aen | | | | |
| Atributos de uso (Au) <ul style="list-style-type: none"> • Au1 • Au2 • . • . • Aun | PMax Au1 PMax Au2 . . PMax Aun | | | | |
| PUNTAJE TOTAL | 100.0 | | | | |

3.9 Valorar resultados

La valoración, que resume un conjunto de niveles calificados, es el paso final del proceso de evaluación del software.

3.10 Documentación

Todo el proceso de evaluación debe estar documentado, indicando nombres y apellidos, cargos, procedencia de las personas que participaron en el proceso de evaluación, especificando las etapas en las que participaron, si es necesario. Este documento deberá ser aprobado por el Jefe de Informática o quien haga sus veces.



GLOSARIO DE TÉRMINOS

Adquiriente

Una organización que adquiere u obtiene un sistema, producto de software o servicio software de un proveedor.

Atributo

Una característica física o abstracta mensurable de una entidad. Los atributos pueden ser internos o externos.

Calidad

Son todas las características de una entidad que forman parte de su habilidad para satisfacer las necesidades propias e implícitas.

Calidad en el empleo

Es la medida en que un producto empleado por usuarios específicos satisface sus necesidades con efectividad, productividad y entera satisfacción para alcanzar objetivos o metas en contextos específicos de su empleo.

Calidad externa

La extensión para la cual un producto satisface necesidades explícitas e implícitas cuando es usado bajo condiciones específicas.

Calidad interna

Es la totalidad de atributos del producto que determinan su habilidad para satisfacer las necesidades propias e implícitas bajo condiciones específicas.

Calificación

La acción de evaluar el valor medido al nivel de calificación adecuado. Utilizado para determinar el nivel de calificación asociado con el software para una característica específica de calidad.

Defecto

Un paso, proceso o definición de dato incorrecto en un programa de computadora.

Desarrollador

Una organización que realiza actividades de desarrollo (incluyendo análisis de los requisitos, diseño y pruebas de aceptación) durante el proceso del ciclo de vida del software.

Escala

Un conjunto de valores con propiedades definidas

Ejemplos de tipos de escalas son: una escala nominal que corresponda a un conjunto de categorías; una escala ordinal que corresponda a un conjunto ordenado de puntos; una escala de intervalo que corresponda a una escala ordenada con puntos equidistantes; y una escala de ratios que no sólo tiene puntos equidistantes sino que



posee el cero absoluto. Las métricas utilizando escalas nominales u ordinales producen datos cualitativos, y las métricas utilizando escalas de intervalos o ratios producen datos cuantitativos.

Falla

La terminación de la capacidad de un producto de realizar una función requerida o su incapacidad para realizarla dentro de límites previamente especificados.

Firmware

El firmware contiene las instrucciones e información acerca del funcionamiento de un dispositivo o hardware, generalmente grabado en un chip. Es el código que rige el comportamiento del mismo.

Indicador

Una medida que se puede utilizar para estimar o para predecir otra medida. Los indicadores pueden emplearse para evaluar los atributos cualitativos del software y para calcular los atributos del proceso de desarrollo. Ambos son valores indirectos e imprecisos de los atributos.

Medición

Actividad que usa la definición de la métrica para producir el valor de una medida.

Medida

Número o categoría asignada a un atributo de una entidad mediante una medición.

Medida directa

Una medida de un atributo que no depende de la medida de ningún otro atributo.

Métrica

Es un método definido de valoración y su escala de valoración.

Las métricas pueden ser internas o externas, directas o indirectas.

Las métricas incluyen métodos para clasificar la data o información cualitativa en diferentes categorías.

Medida externa

Una medida indirecta de un producto derivada de las medidas del comportamiento del sistema del que es parte.

El sistema incluye cualquier hardware, software (ya sea software a medida o software tipo paquete) y usuarios.

El número de fallas encontradas durante las pruebas es una medida externa del número de fallas en el programa, porque el número de fallas es contado durante la operación del programa corriendo en un sistema de cómputo.

Las medidas externas pueden ser usadas para evaluar los atributos de calidad cercanos a los objetivos finales de diseño.

Modelo cualitativo

Es una serie de características y la relación entre las mismas, que conforman la base de los requerimientos cualitativos específicos y la valoración cualitativa.



Módulo de evaluación

Un paquete de tecnología de evaluación para una característica o sub característica de calidad de un software específico. El paquete incluye métodos y técnicas de evaluación, entradas a ser evaluadas, datos a ser medidos y recopilados y procedimientos y herramientas de soporte.

Necesidades implícitas

Necesidades que pueden no haber sido especificadas pero que son necesidades reales cuando la entidad es usada en condiciones particulares.

Necesidades implícitas son necesidades reales, las cuales pueden no haber sido documentadas.

Nivel de calificación

Un punto en la escala ordinal que es utilizado para categorizar una escala de medida. El nivel de calificación habilita al software para ser clasificado de acuerdo con las necesidades explícitas o implícitas. Los niveles de clasificación adecuados pueden ser asociados con las vistas diferentes de calidad, por ejemplo, usuarios, gerentes o desarrolladores.

Producto de software

El conjunto de programas de cómputo, procedimientos, y posible documentación y datos asociados.

Los productos incluyen productos intermedios y productos para los usuarios, como los desarrolladores y personal de soporte.

Producto de software intermedio

Es un producto del proceso de desarrollo del software que se emplea para alimentar una etapa diferente del proceso de desarrollo.

En algunos casos, un producto intermedio puede ser también un producto final.

Proveedor

Una organización que entra a un contrato con el adquirente para el suministro de un sistema, producto de software o servicio de software bajo los términos de dicho contrato.

Servicio

Es una organización que presta servicios de mantenimiento.

Sistema

Una composición integrada que consiste en uno o más procesos, hardware, software, instalaciones y personas, que proveen una capacidad para satisfacer una necesidad establecida o un objetivo.

Software

Todo o parte de los programas, procedimientos, reglas y documentación asociada a un sistema de procesamiento de información.

El software es una creación intelectual que es independiente del medio en el cual fue grabado.



Usuario

Un individuo que utiliza el producto de software para realizar una función específica. Los usuarios pueden incluir operadores, receptores de los resultados del software, desarrolladores o personal de soporte de software.

Valoración

Emplear una métrica para asignar uno de los valores de una escala (el mismo que puede ser un número o categoría) al atributo de una entidad.

La valoración puede ser cualitativa cuando se emplean categorías. Por ejemplo, algunos de los atributos importantes de los productos de software, tales como el lenguaje del programa base (ADA, C, COBOL, etc.) son categorías cualitativas.

Valoración indirecta

Es la valoración de un atributo derivada del valor de uno o más atributos diferentes. La valoración externa de un atributo de un sistema de cómputo (tal como el tiempo de respuesta a la información alimentada por el usuario) es una valoración indirecta de los atributos del software, dado que esta medida se verá influenciada por los atributos del entorno de cómputo, así como por los atributos propios del software.

Valoración interna

Es una valoración del producto en sí, ya sea directa o indirecta.

El número de líneas del código, las valoraciones de complejidad, el número de fallas encontradas durante el proceso y el índice de señales o alertas, son todas las valoraciones internas propias del producto en sí.

Valorar (verbo)

Realizar una valoración o estimación.

Valor (sustantivo)

Es el número o categoría que una entidad le asigna a un atributo al efectuar la valoración.

Valoración Cualitativa

Es una evaluación sistemática del grado o capacidad de una entidad para satisfacer necesidades o requerimientos específicos.

Dichos requerimientos pueden ser formalmente especificados, por ejemplo, por el área de desarrollo de sistemas, cuando el producto se diseña por contrato para un usuario específico, cuando el producto es desarrollado sin un usuario específico, o bien que se trate de necesidades más generales, como cuando un usuario evalúa los productos con propósitos de comparación y selección.

Validación

Confirmación por inspección y provisión de evidencia objetiva de que los requerimientos particulares para un uso específico son alcanzados.

En diseño y desarrollo, la validación está relacionada con el proceso de reexaminación de un producto para determinar la conformidad con las necesidades del usuario.

La validación es realizada normalmente sobre el producto final bajo condiciones operacionales definidas. Puede ser necesaria en las fases iniciales.

“Validado” es utilizado para designar el estado correspondiente.



Verificación

Confirmación por examen y provisión de evidencia objetiva que los requerimientos específicos han sido alcanzados.

En diseño y desarrollo, la verificación está relacionada con el proceso de examinar el resultado de una actividad dada para determinar su conformidad con los requerimientos definidos para dicha actividad.

“Verificado” es utilizado para designar el estado correspondiente.



BIBLIOGRAFÍA

- Norma ISO/IEC 9126-1: 2001 - Software engineering -- Product quality -- Part 1: Quality model.
- Norma ISO/IEC TR 9126-2: 2003 - Software engineering -- Product quality -- Part 2: External metrics.
- Norma ISO/IEC TR 9126-3: 2003 - Software engineering -- Product quality -- Part 3: Internal metrics.
- Norma ISO/IEC 14598-1:1999 - Part 1: General overview.
- Norma ISO/IEC 14598-2:2000 - Part 2: Planning and management.
- Norma ISO/IEC 14598-3:2000 - Part 3: Process for developers.
- Norma ISO/IEC 14598-5:1998 - Part 5: Process for evaluators.