



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®



Especialidad Análisis, diseño y desarrollo de software

D2.2 Atributos de calidad del software

Asesor: M.T.I.C. Leonardo Enriquez

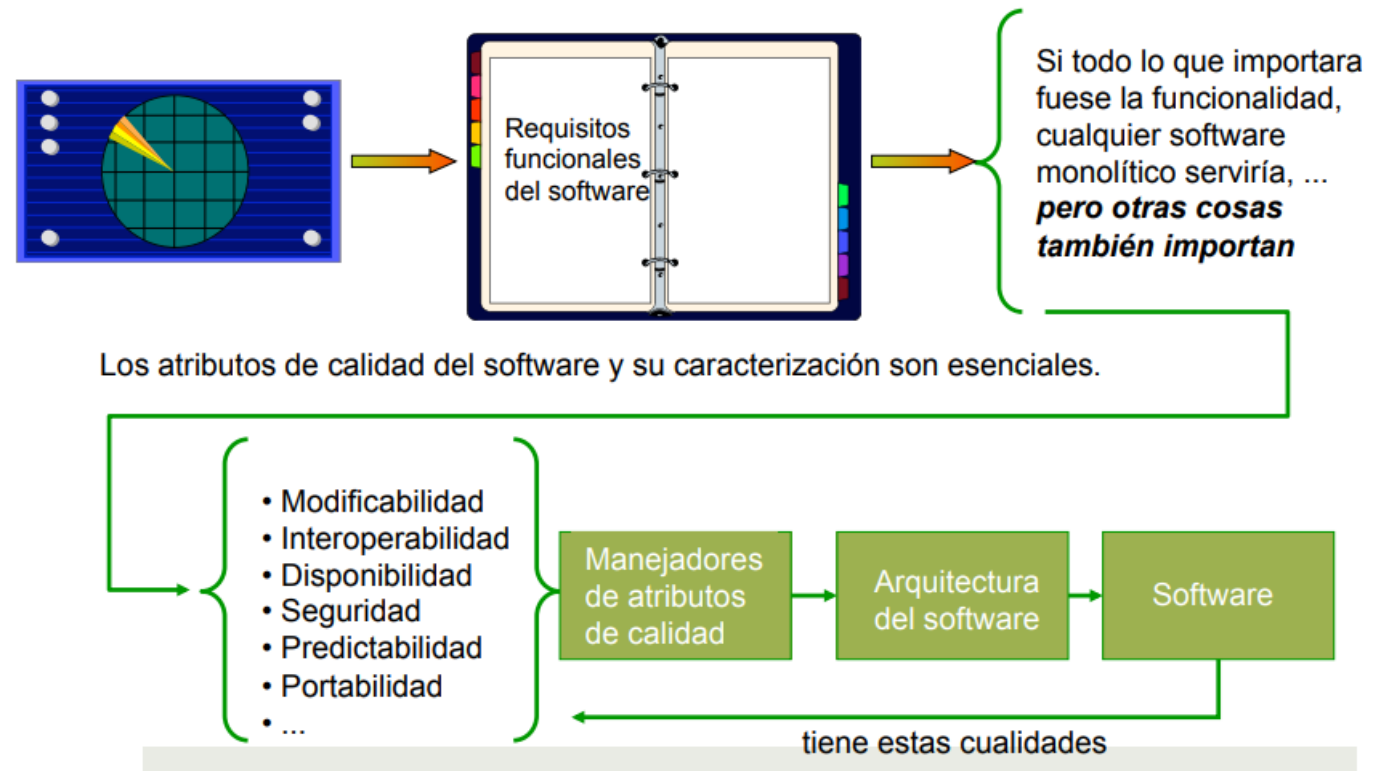
Ingeniero electrónico, sistemas digitales

Atributos de calidad de software

Según Barbacci (1995), la calidad de software se define como el grado en el cual el software posee una combinación deseada de atributos. Tales atributos **son requerimientos adicionales del sistema**, que hacen referencia a características que este debe satisfacer, diferentes a los requerimientos funcionales. Estas características o atributos se conocen como atributos de calidad, los cuales se definen como las propiedades de un servicio que presta el sistema a sus usuarios.

Los atributos de calidad se clasifican en dos categorías

1. **Observables vía ejecución**, son aquellos atributos que se determinan del comportamiento del sistema en tiempo de ejecución.
2. **No observables vía ejecución**, son aquellos atributos que se establecen durante el desarrollo del sistema

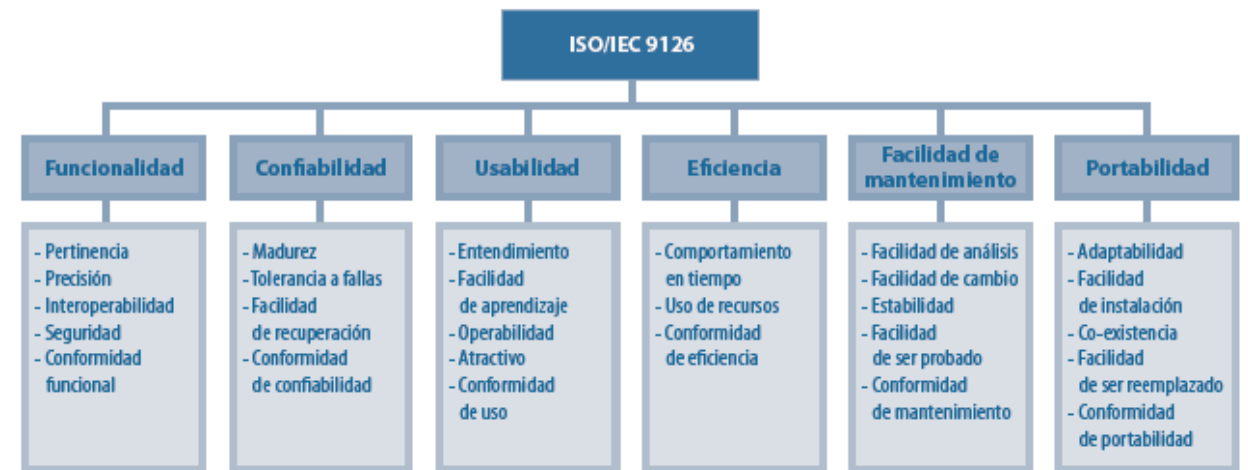


Atributos de calidad de software

Los atributos de calidad especifican características útiles para establecer criterios sobre la calidad del sistema. Actualmente no existe un “listado único” de atributos de calidad relevantes para diseñar la *arquitectura de software*. Sin embargo, estándares para la evaluación de la calidad, como el ISO/IEC 9126 (*International Standard Organization*, 2001) o modelos de calidad como los definidos por McCall’s (Pressman, 2001) o Boehm (Boehm, Brown y Lipow, 1976), pueden ser de utilidad para guiar al arquitecto durante la identificación y la especificación de este tipo de *drivers arquitectónicos*. La figura 2-4 presenta los atributos de calidad que considera el estándar ISO/IEC 9126.

Entre los atributos de calidad, son siete categorías de atributos definidos en Bass, Clements y Kazman, (2012), los cuales son:

1. **Disponibilidad.** Indicador acerca de si el sistema se encuentra en una condición operable cuando requiere ser utilizado.
2. **Seguridad.** Indicador del grado de protección ante usos o accesos inapropiados del sistema.
3. **Desempeño.** Indicador sobre la cantidad de trabajo realizado por el sistema considerando tiempo y recursos.
4. **Facilidad de prueba.** Indicador acerca de la facilidad con la cual se elaboran pruebas efectivas para el sistema.
5. **Modificabilidad.** Indicador referente al costo de realizar cambios en el sistema.
6. **Usabilidad.** Indicador sobre la facilidad con la cual el sistema puede ser utilizado por los usuarios.
10. **Interoperabilidad.** Indicador acerca de la facilidad del sistema para intercambiar información con otros sistemas mediante interfaces.



»Figura 2-4. Atributos de calidad considerados por el estándar ISO/IEC. 9126.

Atributos de calidad Observables vía ejecución

Es importante destacar que tener conocimiento de los atributos observables, no necesariamente implica que se satisfacen los atributos no observables vía ejecución. Por ejemplo, un sistema que satisface todos los requerimientos observables puede o no ser costoso para desarrollar, así como también puede o no ser imposible de modificar.

| Atributo de Calidad | Descripción |
|---------------------------------------|---|
| Disponibilidad (Availability) | Es la medida de disponibilidad del sistema para el uso (Barbacci et al., 1995). |
| Confidencialidad (Confidentiality) | Es la ausencia de acceso no autorizado a la información (Barbacci et al., 1995). |
| Funcionalidad (Functionality) | Habilidad del sistema para realizar el trabajo para el cual fue concebido (Kazman et al., 2001). |
| Desempeño (Performance) | Es el grado en el cual un sistema o componente cumple con sus funciones designadas, dentro de ciertas restricciones dadas, como velocidad, exactitud o uso de memoria. (IEEE 610.12). Según Smith (1993), el desempeño de un sistema se refiere a aspectos temporales del comportamiento del mismo. Se refiere a capacidad de respuesta, ya sea el tiempo requerido para responder a aspectos específicos o el número de eventos procesados en un intervalo de tiempo. Según Bass et al. (1998), se refiere además a la cantidad de comunicación e interacción existente entre los componentes del sistema. |
| Confiabilidad (Reliability) | Es la medida de la habilidad de un sistema a mantenerse operativo a lo largo del tiempo (Barbacci et al., 1995). |
| Seguridad externa (Safety) | Ausencia de consecuencias catastróficas en el ambiente. Es la medida de ausencia de errores que generan pérdidas de información (Barbacci et al., 1995). |
| Seguridad interna (Security) | Es la medida de la habilidad del sistema para resistir a intentos de uso no autorizados y negación del servicio, mientras se sirve a usuarios legítimos (Kazman et al., 2001). |

Atributos de calidad No observables vía ejecución

| Atributo de Calidad | Descripción |
|--|---|
| Configurabilidad (<i>Configurability</i>) | Posibilidad que se otorga a un usuario experto a realizar ciertos cambios al sistema (Bosch et al., 1999). |
| Integrabilidad (<i>Integrability</i>) | Es la medida en que trabajan correctamente componentes del sistema que fueron desarrollados separadamente al ser integrados. (Bass et al. 1998) |
| Integridad (<i>Integrity</i>) | Es la ausencia de alteraciones inapropiadas de la información (Barbacci et al., 1995). |
| Interoperabilidad (<i>Interoperability</i>) | Es la medida de la habilidad de que un grupo de partes del sistema trabajen con otro sistema. Es un tipo especial de <i>integrabilidad</i> (Bass et al. 1998) |
| Modificabilidad (<i>Modifiability</i>) | Es la habilidad de realizar cambios futuros al sistema. (Bosch et al. 1999). |
| Mantenibilidad (<i>Maintainability</i>) | Es la capacidad de someter a un sistema a reparaciones y evolución (Barbacci et al., 1995). Capacidad de modificar el sistema de manera rápida y a bajo costo (Bosch et al. 1999). |
| Portabilidad (<i>Portability</i>) | Es la habilidad del sistema para ser ejecutado en diferentes ambientes de computación. Estos ambientes pueden ser hardware, software o una combinación de los dos (Kazman et al., 2001). |
| Reusabilidad (<i>Reusability</i>) | Es la capacidad de diseñar un sistema de forma tal que su estructura o parte de sus componentes puedan ser reutilizados en futuras aplicaciones (Bass et al. 1998). |
| Escalabilidad (<i>Scalability</i>) | Es el grado con el que se pueden ampliar el diseño arquitectónico, de datos o procedimental (Pressman, 2002). |
| Capacidad de Prueba (<i>Testability</i>) | Es la medida de la facilidad con la que el software, al ser sometido a una serie de pruebas, puede demostrar sus fallas. Es la probabilidad de que, asumiendo que tiene al menos una falla, el software fallará en su próxima ejecución de prueba (Bass et al. 1998). |

- Bosch (2000) establece que los requerimientos de calidad se ven altamente influenciados por la arquitectura del sistema.
- Bass et al. (1998) afirma que la calidad del sistema debe ser considerada en todas las fases de diseño, pero los atributos de calidad se manifiestan de maneras distintas a lo largo de estas fases. De esta forma, establecen que la arquitectura determina ciertos atributos de calidad del sistema, pero existen otros atributos que no dependen directamente de la misma.
- Por ejemplo, la usabilidad de un sistema no está relacionada directamente con la arquitectura del mismo, puesto que los detalles que este atributo envuelve como el uso de botones o radio buttons, pantallas intuitivas, etc, se encuentran casi siempre encapsulados en un componente simple.

Modelos de calidad

En la practica, los modelos de calidad resultan de utilidad para la predicción de confiabilidad y en la gerencia de calidad durante el proceso de desarrollo, así como para efectuar la medición del nivel de complejidad de un sistema de software (Kan et al, 1994).

Pressman (2002) indica que los factores que afectan a la calidad del software no cambian, por lo que resulta útil el estudio de los modelos de calidad que han sido propuesto en este sentido. Dado que los factores de calidad siguen siendo desde el año 70 validos, se mostraran los mas importantes y adaptados para la arquitectura de software propuesta por Losavia et al. (2003)

- Modelo de McCall (1977), describe la calidad como un concepto elaborado mediante relaciones jerárquicas entre factores de calidad, en base a criterios y métricas de calidad.
- Modelo Dromery (1996), propone un marco de referencia o metamodelo para la construcción de modelos de calidad, basado en como las propiedad medibles de un producto de software puede afectar los atributos de calidad generales, como por ejemplo, confiabilidad y mantenibilidad.
- Modelo FURPS (1987), el modelo McCall ha servido de base para los modelos de calidad posteriores y este es el caso del modelo FURPS, producto del desarrollo de Hewlett-Packard.
- ISO/IEC 9126 (1991), este estándar ha sido desarrollado en un intento de identificar los atributos clave de calidad para un producto de software (Pressman, 2002).
- ISO/IEC 9126

Modelo de calidad McCall

Modelo de McCall

Este modelo se enfoca sistemáticamente en cuantificar la calidad a través de las siguientes fase:

- Determinación de los factores que influyen sobre la calidad del software.
- Identificación de los criterios para juzgar cada factor.
- Definición de las métricas de los criterios y establecimiento de una función de normalización que define la relación entre las métricas de cada criterio y los factores correspondientes.
- Evaluación de las métricas.
- Correlación de las métricas a un conjunto de guías que cualquier equipo de desarrollo podría seguir.
- Desarrollo de las recomendaciones para la colección de métricas.

En el modelo de McCall, los factores de calidad se concentran en tres aspectos importantes de un producto de software: características operativas, capacidad de cambios y adaptabilidad a nuevos entornos.

La tabla muestra para el modelo McCall, los factores de calidad y sus criterios asociados. En ella se observa que algunos de los criterios de calidad son compartidos por más de un factor.

| Factor | Criterio |
|---------------------|--|
| Correctitud | ✓ Rastreabilidad ✓ Completitud ✓ Consistencia |
| Confiabilidad | ✓ Consistencia ✓ Exactitud ✓ Tolerancia a fallas |
| Eficiencia | ✓ Eficiencia de ejecución ✓ Eficiencia de almacenamiento |
| Integridad | ✓ Control de acceso ✓ Auditoría de acceso |
| Usabilidad | ✓ Operabilidad ✓ Entrenamiento ✓ Comunicación |
| Mantenibilidad | ✓ Simplicidad ✓ Concreción |
| Capacidad de Prueba | ✓ Simplicidad ✓ Instrumentación ✓ Auto-descriptividad ✓ Modularidad |
| Flexibilidad | ✓ Auto-descriptividad ✓ Capacidad de expansión ✓ Generalidad ✓ Modularidad |
| Portabilidad | ✓ Auto-descriptividad ✓ Independencia del sistema ✓ Independencia de máquina |
| Reusabilidad | ✓ Auto-descriptividad ✓ Generalidad ✓ Modularidad ✓ Independencia del sistema ✓ Independencia de máquina |
| Interoperabilidad | ✓ Modularidad ✓ Similitud de comunicación ✓ Similitud de datos. |

Modelo McCall , criterios asociados a los factores de calidad
Fuente: McCall, 1977

Modelo de calidad Dromey

Modelo de Dromey

- Este modelo plantea como conectar propiedad del producto con los atributos de calidad de alto nivel, sugiriendo el uso de cuatro categorías que implican propiedad de calidad que son: correctitud, internas, contextuales y descriptivas.
- El proceso de construcción de modelos de calidad propuesto por Dromey (1996) consta de 5 pasos, basados en las propiedad mencionadas. Los pasos del marco de referencia propuesto son:
- Especificación de los atributos de calidad de alto nivel (por ejemplo, confiabilidad, mantenibilidad)
- Determinación de los distintos componentes del producto a un apropiado nivel de detalle (por ejemplo, paquetes, subrutinas, declaraciones)
- Para cada componente, determinación y categorización de sus implicaciones mas importantes de calidad.
- Proporción de enlaces que relacionan las propiedades implícitas a los atributos de calidad, o alternatively, uso de enlaces de las cuatro categorías de atributos propuestas.
- Iteración sobre los pasos anteriores, utilizando un proceso de evaluación y refinamiento.

| Propiedades del producto | Atributos de Calidad |
|--------------------------|---|
| Correctitud | ✓ Funcionalidad ✓ Confiabilidad |
| Internas | ✓ Mantenibilidad ✓ Eficiencia ✓ Confiabilidad |
| Contextuales | ✓ Mantenibilidad ✓ Reusabilidad ✓ Portabilidad ✓ Confiabilidad |
| Descriptivas | ✓ Mantenibilidad ✓ Reusabilidad ✓ Portabilidad ✓ Usabilidad |

Modelo de Dromey, relación entre propiedades del producto y atributos de calidad

Fuente: Dromey, 1996

Modelo de calidad FURPS

Modelo FURPS

- Este modelo desarrolla un conjunto de factores de calidad de software, bajo el acrónimo de FURPS: funcionalidad, usabilidad, confiabilidad, desempeño y capacidad de soporte.
- La tabla presenta la clasificación de los atributos de calidad que se incluyen en el modelo, junto con las características asociadas a cada uno (Pressman, 2002).
- Este modelo incluye además factores de calidad y atributos, restricciones de diseño y requerimientos de implementación, físicos y de interfaz. Las restricciones de diseño especifican o restringen el diseño del sistema.
- Los requerimientos de implementación especifican o restringen la codificación o construcción de un sistema (ejemplo, estándares requeridos, lenguajes, políticas).
- Los requerimientos de interfaz especifican el comportamiento de los elementos externos con los que el sistema debe interactuar.
- Los requerimientos físicos especifican ciertas propiedades que el sistema debe poseer, en términos de materiales, forma, peso, tamaño (por ejemplo, requisitos de hardware, configuración de red).

| Factor de Calidad | Atributos |
|----------------------|--|
| Funcionalidad | ✓ Características y capacidades del programa ✓ Generalidad de las funciones ✓ Seguridad del sistema |
| Facilidad de uso | ✓ Factores humanos ✓ Factores estéticos ✓ Consistencia de la interfaz ✓ Documentación |
| Confiabilidad | ✓ Frecuencia y severidad de las fallas ✓ Exactitud de las salidas ✓ Tiempo medio de fallos ✓ Capacidad de recuperación ante fallas ✓ Capacidad de predicción |
| Rendimiento | ✓ Velocidad del procesamiento ✓ Tiempo de respuesta ✓ Consumo de recursos ✓ Rendimiento efectivo total ✓ Eficacia |
| Capacidad de Soporte | ✓ Extensibilidad ✓ Adaptabilidad ✓ Capacidad de pruebas ✓ Capacidad de configuración ✓ Compatibilidad ✓ Requisitos de instalación |

Modelo FURPS , atributos de calidad
Fuente: Pressman, 2002

Modelo de calidad ISO/IEC 9126

Modelo ISO/IEC 9126

- Este estándar es una simplificación del Modelo de McCall (Losavia, 2003), e identifica seis características básicas de calidad que pueden estar presentes en cualquier producto de software.
- El estándar provee una descomposición de las características en sub-características, que se demuestran en la tabla.
- Es interesante destacar que los factores de calidad que contempla este estándar no son necesariamente usados para mediciones directas (Pressman, 2002) pero proveen una valiosa base para medidas indirectas, y una excelente lista para determinar la calidad de un sistema.

| Característica | Subcaracterística |
|----------------|------------------------------|
| Funcionalidad | ✓ Adecuación |
| | ✓ Exactitud |
| | ✓ Interoperabilidad |
| | ✓ Seguridad |
| Confiabilidad | ✓ Madurez |
| | ✓ Tolerancia a fallas |
| | ✓ Recuperabilidad |
| Usabilidad | ✓ Entendibilidad |
| | ✓ Capacidad de aprendizaje |
| | ✓ Operabilidad |
| Eficiencia | ✓ Comportamiento en tiempo |
| | ✓ Comportamiento de recursos |
| Mantenibilidad | ✓ Analizabilidad |
| | ✓ Modificabilidad |
| | ✓ Estabilidad |
| | ✓ Capacidad de pruebas |
| Portabilidad | ✓ Adaptabilidad |
| | ✓ Instalabilidad |
| | ✓ Reemplazabilidad |

Modelo ISO/IEC 9126 , características y sub-características de calidad

Fuente: Pressman, 2002

Modelo de calidad ISO/IEC 9126 adaptado para arquitectura de software

Modelo ISO/IEC 9126 adaptado para arquitectura de software

- Losavia et al (2003) propone una adaptación del modelo ISO/IEC 9126 de calidad de software para efectos de la evaluación de la arquitectura de software. El modelo se basa en los atributos de calidad que se relacionan directamente con la arquitectura: **funcionalidad, confiabilidad, eficiencia, mantenibilidad y portabilidad.**
- La característica de usabilidad propuesta por el modelo ISO/IEC 9126 puede ser refinada para obtener atributos que se relacionan con los componentes de la interfaz con el usuario. Dado que estos componentes son independientes de la arquitectura, no son considerados en la adaptación del modelo.
- La tabla presenta los atributos de calidad planteados por Losavio et al, (2003), que poseen subcaracterísticas asociadas con elementos de tipo arquitectónico.

| Característica | Subcaracterística | Elementos de tipo arquitectónico |
|----------------|-------------------------|---|
| Funcionalidad | Adecuación | Refinamiento de los diagramas de secuencia |
| | Exactitud | Identificación de los componentes con las funciones responsables de los cálculos |
| | Interoperabilidad | Identificación de conectores de comunicación con sistemas externos |
| | Seguridad | Mecanismos o dispositivos que realizan explícitamente la tarea |
| Confiabilidad | Tolerancia a fallas | Existencia de mecanismos o dispositivos de software para manejar excepciones |
| | Recuperabilidad | Existencia de mecanismos o dispositivos de software para reestablecer el nivel de desempeño y recuperar datos |
| Eficiencia | Desempeño | Componentes involucrados en un flujo de ejecución para una funcionalidad |
| | Utilización de recursos | Relación de los componentes en términos de espacio y tiempo |
| Mantenibilidad | Acoplamiento | Interacciones entre componentes |
| | Modularidad | Número de componentes que dependen de un componente |
| Portabilidad | Adaptabilidad | Presencia de mecanismos de adaptación |
| | Instalabilidad | Presencia de mecanismos de instalación |
| | Coexistencia | Presencia de mecanismos que faciliten la coexistencia |
| | Reemplazabilidad | Lista de componentes reemplazables para cada componente |

Atributos de calidad planteados por Losavio (2003), que posee sub-características asociadas con elementos de tipo arquitectónico.

Bibliografía

Pressman, R. S. (2010). Ingeniería de Software, Un enfoque practico Séptima Edición. Ciudad de México: Mc Graw Hill.

Sommerville. (2011). Ingeniería de Software 9 Edición. Estado de México: Pearson.

UNID Universidad Interamericana para el desarrollo. (2018). Ingeniería de software. Ciudad de México: UNID.

Cervantes, H. (2016). Arquitectura de Software, Conceptos y ciclo de desarrollo. Ciudad de México: Cengage Learning.