

Testing Manual

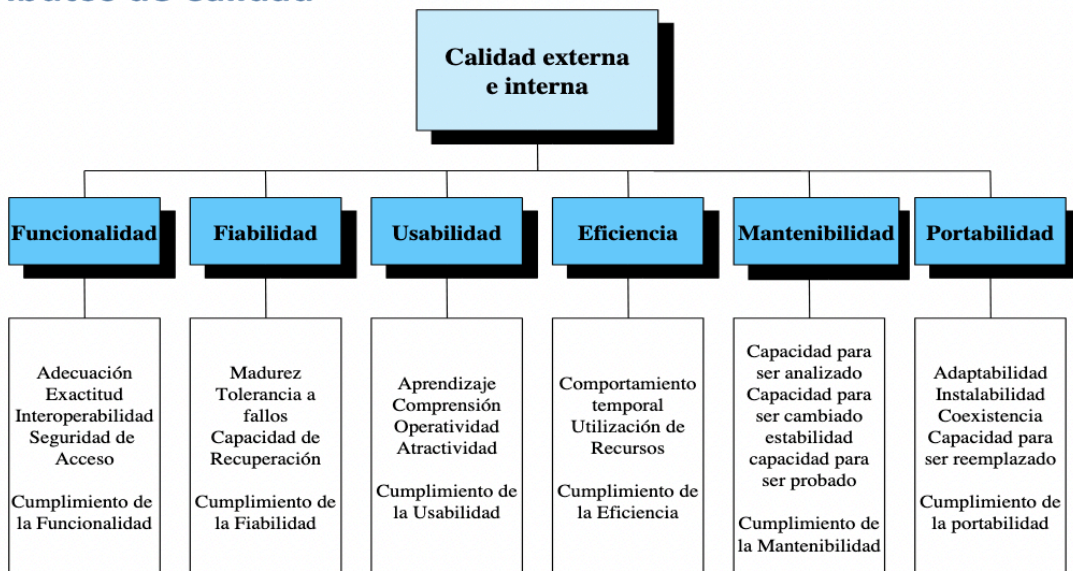
Los testers se basan en una combinación de habilidades técnicas, comprensión del negocio y colaboración efectiva para garantizar la calidad del software. Su objetivo principal es asegurarse de que el software cumpla con los requerimientos y expectativas del usuario final.

En búsqueda de la calidad...

Como hemos explorado anteriormente, la búsqueda constante de calidad es esencial para el equipo de software. El equipo de testing desempeña un papel crucial en asegurar esta calidad, fundamentándose en los principios y objetivos inherentes al proceso de pruebas. Además, emplean normas específicas para evaluar la calidad del software, de las cuales mencionaremos dos destacadas:

- **CALIDAD ISO / IEC 9126**

Atributos de Calidad



💡 A continuación te dejamos el mismo gráfico pero en su versión original.

Es importante que puedas aprender vocabulario específico en inglés ya que -como mencionamos en el curso anterior- es el idioma predominante en la industria IT.

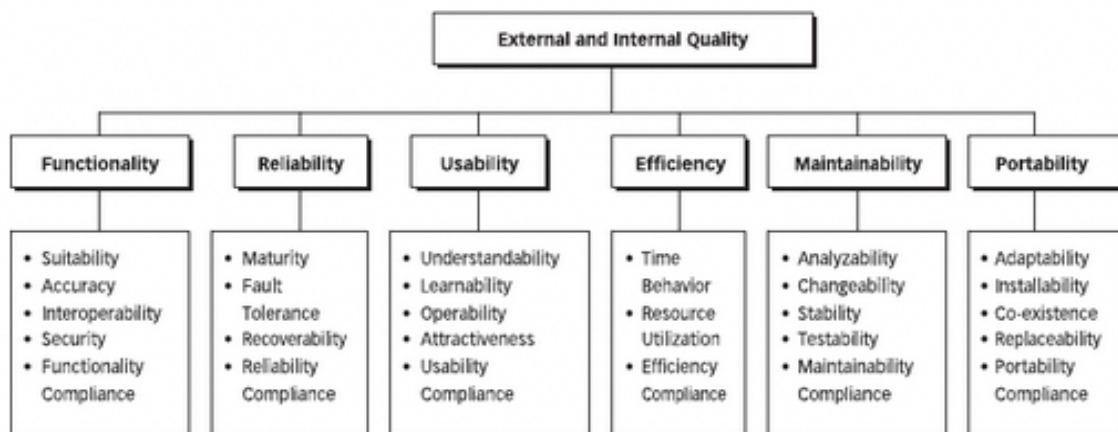


Figure 10.2. Quality model for External and Internal Quality: characteristics and sub-characteristics—ISO 9126-1. This figure was adopted from Figure 4 (p.7) of ISO/IEC 9126-1:2001(E). This figure is not to be considered an official ISO figure nor was it authorized by ISO. Copies of ISO/IEC 9126-1:2001(E) can be purchased from ANSI at <http://webstore ansi.org>.

Puedes profundizar más leyendo el siguiente [artículo](#).

- **ISO / IEC 25010 SQuaRE — System and Software Quality Requirements and Evaluation.**



Para conocer más sobre esta norma, te invitamos a leer el siguiente [artículo](#)

Ambas normas abarcan los mismos aspectos. Sin embargo existen algunas diferencias:

- En el gráfico **ISO 9126**, seguridad y compatibilidad (e interoperabilidad) están considerado *dentro* de la rama de Funcionalidad
- En el gráfico **ISO 25010** esos mismos aspectos aparecen *separados* y algo más desglosados.

Es importante tener en cuenta estas normas referidas a la calidad de software a la hora de desarrollar las tareas de testing: desde el análisis de los requerimientos hasta el diseño del plan de pruebas.

¿Cómo se organiza el área de testing?

Dependiendo la empresa y sus recursos disponibles, te dejamos aquí una división tradicional del área, con algunos puntos básicos a tomar en cuenta en su ejecución.

EMPRESAS SIN ÁREA DE TESTING:

- La responsabilidad cae sobre el área de desarrollo
- Deben definir cuantos aspectos de la calidad quieren / pueden cubrir en los planes de pruebas.
- Pueden existir roles especializados en automatizar las pruebas.
- Existen pruebas de integración que buscan garantizar que los diferentes componentes del software trabajen juntos de manera eficiente.

EMPRESAS CON ÁREA DE TESTING:

- Testing de Seguridad: Dedicado a la evaluación sistemática de un sistema o aplicación para identificar y corregir vulnerabilidades de seguridad
- Testing de Performance: Dedicado a evaluar rendimiento y capacidad de respuesta.
- Testing de Usabilidad: Dedicado a la evaluación de la facilidad con la que los usuarios pueden interactuar con un sistema o aplicación
- Testing Funcional: Pruebas que se centran en verificar si una aplicación o sistema cumple con los requisitos especificados

Cada empresa encuentra y diseña el departamento ajustándolo a sus propias necesidades, priorizando su funcionalidad en relación a los objetivos previamente establecidos.

Testing Dinámico y Estático

Profundizando en el enfoque del testing estático y dinámico, como hemos explorado previamente, estos dos enfoques son esenciales en el proceso de aseguramiento de la calidad del software, cada uno desempeñando un papel único con propósitos y métodos específicos. Ahora, nos sumergiremos más detenidamente en estas perspectivas, las cuales, en conjunto, proporcionan herramientas cruciales para evaluar y mejorar la calidad del software. A continuación, se ofrece una explicación detallada de ambos enfoques:

Testing Estático

El testing estático se centra en la revisión y análisis del código y la documentación sin ejecutar el programa. Se lleva a cabo durante las primeras etapas del ciclo de vida del desarrollo de software.

Actividades Principales:

- 1. Revisión de Código:** Los desarrolladores y otros miembros del equipo revisan el código fuente para identificar errores, inconsistencias y posibles mejoras.
- 2. Análisis Estático:** Se examina la documentación, como especificaciones y diseños, para asegurarse de que sea coherente y cumpla con los estándares establecidos.
- 3. Inspecciones y Revisiones:** Sesiones formales donde se revisan el código, los diseños y la documentación para encontrar defectos y mejorar la calidad del software.

Ventajas:

- Identificación temprana de errores.
- Menor costo de corrección.
- Mejora la comprensión del código y de los requisitos.

Limitaciones:

- No garantiza la ausencia total de errores.
- No evalúa el comportamiento en tiempo de ejecución.

Testing Dinámico:

El testing dinámico implica la ejecución real del software y la observación de su comportamiento durante la ejecución. Se realiza principalmente durante las etapas de desarrollo avanzado.

Actividades Principales:

1. **Diseño de Casos de Prueba:** Creación de casos de prueba basados en requisitos para evaluar el comportamiento del software.
2. **Ejecución de Pruebas:** Los casos de prueba se ejecutan y se observa cómo responde el software a diferentes entradas y situaciones.
3. **Análisis de Resultados:** Evaluación de los resultados de las pruebas para identificar defectos y áreas de mejora.

Ventajas:

- Evalúa el comportamiento real del software.
- Identificación de defectos que pueden no ser evidentes sin la ejecución del programa.

Limitaciones:

- Costos asociados con la creación y ejecución de pruebas.
- Puede ser más tardío en la identificación de errores en comparación con el testing estático.

Comparación:

- Complementariedad: Ambos tipos de testing son complementarios y se utilizan de manera conjunta en un enfoque integral de aseguramiento de calidad.
- Fases de Aplicación: El testing estático generalmente se aplica en las primeras fases del desarrollo, mientras que el testing dinámico se realiza en fases más avanzadas cuando el software está en ejecución.
- Costos y Beneficios: El testing estático tiende a ser más económico y eficiente en la detección temprana de errores, mientras que el testing

dinámico es crucial para evaluar el comportamiento real del software y su calidad general.

En conclusión, ambos enfoques son esenciales para garantizar un software de alta calidad, ya que cada uno aborda diferentes aspectos del proceso de desarrollo y evaluación.

A continuación, te invitamos a ver el siguiente video para concluir el analisis estatico y dinamico del testing

👉 "Testing estático vs. Testing dinámico" 🎥 [\[video\]](#)