

SERVICIO NACIONAL DE APRENDIZAJE SENA
CENTRO DE SERVICIOS FINANCIEROS

TECNOLOGO EN ANALISIS Y DESARROLLO DE
SOFTWARE

FICHA: 2627063

INSTRUCTOR: ANDRES RUBIANO CUCARIAN

APRENDIZ:

Ana Gabriela Guerrero Martínez

**Evidencia: Documento con las verificaciones de condiciones de
calidad del producto de software ajustado. GA11-220501098-AA3-
EV02**

2024

INTRODUCCIÓN

Este trabajo expondremos las técnicas y herramientas utilizadas en estas verificaciones, sino que también destaca la importancia de un enfoque sistemático y riguroso para garantizar la calidad del producto de software. Al final, la adopción de estas prácticas de verificación contribuye significativamente al éxito del software.

DESCRIPCIÓN DEL PROYECTO

Verificación de condiciones de calidad en proceso de ajuste

Las verificaciones de condiciones de calidad del producto de software ajustado son esenciales para asegurar que el software cumple con los estándares requeridos y satisface las expectativas de los usuarios. Aquí se describen varios enfoques y técnicas para realizar estas verificaciones:

1. Pruebas Funcionales

- **Pruebas Unitarias**

Objetivo: Verificar que las unidades individuales de código (módulos, funciones) funcionan según lo esperado.

Método: Cada unidad se prueba de forma aislada utilizando casos de prueba específicos.

Herramientas: JUnit

- **Pruebas de Integración**

Objetivo: Asegurar que los módulos o componentes integrados funcionen correctamente juntos.

Método: Se prueban las interfaces y la interacción entre módulos.

Herramientas: Postman

- **Pruebas del Sistema**

Objetivo: Validar el comportamiento del sistema completo contra los requisitos funcionales.

Método: Pruebas end-to-end que simulan casos de uso reales.

- Pruebas de Aceptación

Objetivo: Confirmar que el software cumple con las expectativas del cliente o usuario final.

Método: Ejecución de pruebas basadas en criterios de aceptación definidos por los stakeholders.

2. Pruebas No Funcionales

- Pruebas de Rendimiento

Objetivo: Evaluar el comportamiento del sistema bajo diferentes condiciones de carga y estrés.

Método: Simulación de múltiples usuarios y análisis del tiempo de respuesta, uso de recursos y estabilidad.

- Pruebas de Usabilidad

Objetivo: Medir la facilidad de uso y la experiencia del usuario.

Método: Observación de usuarios reales mientras interactúan con el sistema y recopilación de feedback.

- Pruebas de Seguridad

Objetivo: Identificar vulnerabilidades y asegurar que el sistema protege los datos y resiste ataques.

Método: Escaneo de vulnerabilidades, pruebas de penetración y revisión de código de seguridad.

- **Pruebas de Compatibilidad**

Objetivo: Asegurar que el software funcione correctamente en diferentes entornos.

Método: Pruebas en diversos navegadores, sistemas operativos y dispositivos.

3. Pruebas de Mantenimiento

- **Pruebas de Regresión**

Objetivo: Verificar que los cambios o ajustes no introduzcan nuevos errores en el sistema.

Método: Re ejecución de casos de prueba existentes después de cada cambio en el código.

Herramientas: JUnit.

- **Pruebas de Revalidación**

Objetivo: Confirmar que los errores identificados y corregidos hayan sido solucionados adecuadamente.

Método: Ejecución de casos de prueba específicos que originalmente fallaron. **Herramientas:** Los mismos utilizados para las pruebas iniciales.

4. Evaluación de Resultados y Mejora Continua

- **Análisis de Resultados de Pruebas**

Método: Revisar los resultados de todas las pruebas para identificar patrones y áreas comunes de fallos.

- **Retroalimentación y Ajustes**

Método: Recoger retroalimentación de usuarios y equipo de desarrollo, y realizar ajustes iterativos.

Calidad basada en el proceso de ajustes

MODELOS BASADO EN PROCESO CMM (Capability Maturity Model) y CMMI (CMM Integrated)

Datos Importantes:

Creado como un marco (framework) para varias disciplinas relacionadas:

- Ingeniería de sistemas
- Ingeniería de software
- Desarrollo integrado de productos y procesos

Ante el aumento de métodos de estimar capacidad y de evaluar procesos, se necesitaba un método más manejable a nivel de proceso, de proyecto. Además, se requiere poder comparar evaluaciones.

Propósitos:

Se aplica a mejorar los procesos y a determinar la capacidad aplicable a diferentes dominios, necesidades y tamaño de organización no supone estructura organizacional, filosofía administrativa, modelo e ciclo de vida, tecnologías de software o método de desarrollo usa criterios objetivos y prefiere cuantitativos salida en forma de perfiles comparables.

Etapas:

- Preparación: alcance, metas del negocio, procesos a evaluar, instancias de proceso
- Recolección de datos: expertos
- Análisis de datos, asignar niveles, preparar salida

CALIDAD BASADA EN PRODUCTO o ISO/IEC 9126

Esta Norma permite especificar y evaluar la calidad del software desde distintas perspectivas, las cuales están asociadas a la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad, y auditoria del software. Puede ser usada por desarrolladores, evaluadores independientes y grupos de aseguramiento de la calidad responsables de especificar y evaluar la calidad del software. La evaluación de los productos de software que satisfacen las necesidades de calidad del software es uno de los procesos del ciclo de vida de desarrollo del software. La calidad del producto de software puede ser evaluada por medio de la medición de atributos internos, externos o a través de la calidad en uso. El objetivo es que el producto tenga el efecto requerido en un contexto particular de uso.

- Funcionalidad

Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

1. Adecuación

2. Exactitud

3. Interoperabilidad

4. Seguridad

5. Cumplimiento funcional

- Confiabilidad

Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

1. Madurez

2. Recuperabilidad

3. Tolerancia a fallos

4. Cumplimiento de Fiabilidad

- Usabilidad

Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

1. Aprendizaje

2. Comprensión

3. Operatividad

4. Atractividad

- Eficiencia

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesarios bajo condiciones establecidas.

1. Comportamiento en el tiempo

2. Comportamiento de recursos

- Mantenibilidad

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

1. Estabilidad

2. Facilidad de cambio análisis

3. Facilidad de pruebas

- Portabilidad

Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra.

1. Capacidad de instalación

2. Capacidad de ejecución

Definición de Estándares de Calidad

- **Documentación de Requisitos:**

Hay que asegurar que todos los requisitos estén claramente definidos y documentados. Realizar revisiones y aprobaciones formales de los documentos de requisitos.

- **Diseño de Arquitectura y Interfaz:**

Seguir buenas prácticas de diseño y arquitectura de software. Utilizar patrones de diseño reconocidos y estándares de usabilidad. Revisar y aprobar los diseños antes de comenzar el desarrollo.

1. Planificación de la Calidad

- **Plan de Gestión de la Calidad:**

Desarrollar un plan de calidad que defina los objetivos de calidad, métricas, y actividades de aseguramiento de la calidad.

- **Capacitación del Equipo:**

Proporcionar capacitación sobre estándares de calidad y mejores prácticas de desarrollo. Asegurar que todos los miembros del equipo estén familiarizados con las herramientas y procesos utilizados.

2. Control de Calidad durante el Desarrollo

- **Revisiones de Código:**

Implementar revisiones de código regulares para asegurar que el código cumpla con los estándares de calidad.

- **Pruebas Unitarias:**

Desarrollar y ejecutar pruebas unitarias para cada componente del sistema.

- **Pruebas de Integración:**

Realizar pruebas de integración para asegurar que los componentes del sistema funcionen correctamente juntos.

3. Control de Calidad durante las Pruebas

- **Pruebas Funcionales:**

Asegurar que todas las funcionalidades especificadas en los requisitos se prueben exhaustivamente.

- **Pruebas de Rendimiento:**

Realizar pruebas de rendimiento para asegurar que el sistema pueda manejar la carga esperada sin degradación del rendimiento.

- **Pruebas de Usabilidad:**

Realizar pruebas de usabilidad con usuarios finales para asegurar que el sistema sea fácil de usar y cumpla con las expectativas de los usuarios.

4. Revisión y Mejora Continua

- **Revisiones de Proceso:** Realizar revisiones periódicas del proceso de desarrollo para identificar áreas de mejora. Implementar cambios en el proceso basados en las lecciones aprendidas y mejores prácticas.

- Auditorías de Calidad: Llevar a cabo auditorías de calidad para asegurar que se sigan los procesos y estándares de calidad.

- **Retroalimentación y Mejora Continua:**

Recoger retroalimentación de los usuarios finales y del equipo de desarrollo sobre la calidad del sistema y del proceso.

5. Métricas de Calidad

- **Métricas de Producto:**

Tasa de defectos (número de defectos por línea de código). Cobertura de pruebas unitarias y de integración.

Rendimiento del sistema bajo diferentes cargas.

- **Métricas de Proceso:**

Tiempo medio de resolución de defectos. Adherencia a los cronogramas del proyecto. Número de revisiones y auditorías completadas.

6. Documentación de Calidad

- **Registro de Requisitos y Cambios:**

Mantener un registro detallado de todos los requisitos y cambios aprobados.

Asegurar que los cambios se documenten y se comuniquen adecuadamente a todos los interesados.

- **Documentación de Pruebas:**

Documentar todos los casos de prueba, resultados de pruebas y defectos encontrados. Mantener un historial de todas las pruebas realizadas y su estado.

- **Informes de Calidad:**

Generar informes periódicos de calidad que resuman el estado de la calidad del producto y del proceso.

Compartir estos informes con todos los interesados para asegurar la transparencia y la toma de decisiones informada

CONCLUSIÒN

Las verificaciones de condiciones de calidad del producto de software ajustado son fundamentales para garantizar la excelencia del software entregado. A través de una combinación de pruebas funcionales y no funcionales, como pruebas unitarias, de integración, de rendimiento y de seguridad, se asegura que el software no solo cumple con los requisitos especificados, sino que también es robusto, eficiente y segura.