IEEE 1008- Standard for Software Unit Testing

OBJETIVOS

- El objetivo principal del estándar es especificar un enfoque estándar para las pruebas de unidad del software que pueda ser usado como base firme para para la práctica de la ingeniería de software
- Describir los conceptos de la ingeniería de software y las suposiciones de las pruebas en las cuales se basa el enfoque del estándar
- Proporcionar directrices y fuentes de información para asistir con la implementación y uso del enfoque del estándar de pruebas de unidad

MOTIVACIÓN

- Ayuda a la comunicación al proporcionar una descomposición estándar del proceso de prueba de la unidad.

AUDIENCIA

- Testers de unidad y supervisores de testers de unidad
- Fue desarrollado para asistir a aquellos quienes proporcionan entradas, realizan, supervisan, monitorean y evalúan las pruebas de unidad

RELACIÓN CON OTROS ESTÁNDARES DE LA INGENIERÍA DE SOFTWARE

ANSI/IEEE Std 829-1983: Estándar para la documentación de las pruebas del software:

- Describe la información básica necesaria y resultados de las pruebas de software

TERMINOLOGÍA

- Es consistente con el estándar ANSI/IEEE 729-1983. Glosario estándar de terminología de Ingeniería de Software.
- Las pruebas de unidad mencionadas en este estándar son un caso específico del elemento de prueba referido en el ANSI/IEEE 829-1983

Para evitar inconsistencias, el glosario es revisado, sus definiciones no son repetidas en el estándar.

PERSPECTIVA GENERAL

Está compuesto de tres fases, divididas en ocho actividades básicas:

1) Realizar el plan de pruebas
    a) Planear el enfoque general, recursos y calendario
    b) Determinar las características a ser probadas
    c) Refinar el plan general
2) Hacer el conjunto de pruebas
    a) Diseñar el conjunto de pruebas
    b) Implementar el plan redefinido y diseño
3) Medir la prueba de unidad
    a) Ejecutar los procedimientos de prueba
    b) Verificar la terminación
    c) Evaluar el esfuerzo de pruebas y unidad

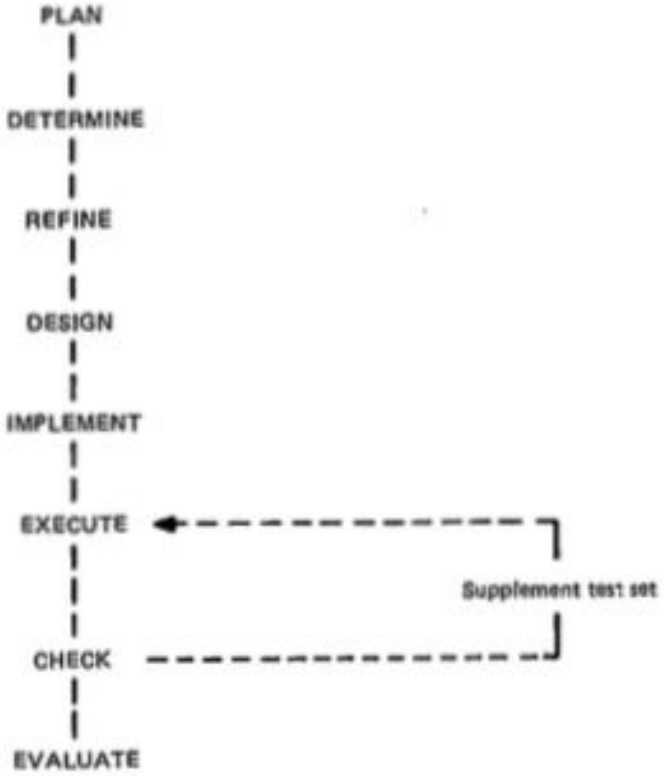


Fig 1 — Unit Testing Activities

HISTORIA

- Se anunció la formación de un grupo de trabajo en la prensa técnica a fines del 1982
- Se comenzó a trabajar en el estándar en febrero de 1983
- La solicitud del proyecto fue aprobada por el Consejo de Normas del IEEE el 23 de junio de 1983
- En marzo de 1985 se produjo el borrador sometido a votación.
- Un total de más de 90 personas contribuyeron al desarrollo inicial de esta norma.

1. Alcance y referencias

DENTRO DEL ALCANCE

- La prueba de la unidad de software es un proceso que incluye el rendimiento de la planificación de prueba, la adquisición de un conjunto de prueba y la medición de una unidad de prueba según sus requisitos.
- El estándar define un enfoque integrado a las pruebas de unidad sistemáticas y documentadas.
- El enfoque usa diseño de unidad e información de implementación de unidad, además de los requerimientos de unidad, para determinar la completitud de las pruebas
- El estándar requiere la realización de cada actividad. Para cada tarea dentro de una actividad, este estándar requiere que se realice la tarea o que los resultados previos estén disponibles y se vuelvan a verificar.
- La planificación de la prueba de la unidad general debe realizarse durante la planificación general de la prueba.
- Esta norma se puede aplicar a las pruebas unitarias de cualquier software o firmware digital.

FUERA DEL ALCANCE

- Los resultados de algunas tareas generales se aplican a todos los niveles de prueba
    - Se deben identificar restricciones de seguridad y privacidad.
- Tales tareas no se consideran parte del proceso de pruebas unitarias, aunque sí las afectan directamente.
- Se identifica una necesidad de información de análisis de fallas y corrección de fallas de software, pero no especifica un proceso de depuración de software.
- No aborda otros componentes de un proceso integral de verificación y validación de unidades, como revisiones , análisis estático o análisis formal.
- Este estándar no requiere el uso de instalaciones o herramientas de prueba específicas.
- Esta norma no implica ninguna metodología particular para el control de la documentación, la gestión de la configuración, la garantía de calidad o la gestión del proceso de prueba

REFERENCIAS

Esta norma se usará junto con las siguientes publicaciones.

- ANSI/IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology.
- ANSI/IEEE Std 829-1983, IEEE Standard for Software Test Documentation.

2. DEFINICIONES

Se definen los términos clave usados en este estándar

- **Lenguaje de programación no procedimental:** Un lenguaje de programación de computadora utilizado para expresar los parámetros de un problema en lugar de los pasos en una solución.
- **Lenguaje de programación procedimental:** Un lenguaje de programación de computadora utilizado para expresar la secuencia de operaciones que realizará una computadora.
- **Característica del software**: Un rasgo, calidad o propiedad inherente, posiblemente accidental del software. Una característica de software especificada o implícita en la documentación de requisitos
- **Incidente de prueba de software:** Cualquier evento que ocurra durante la ejecución de una prueba de software que requiera investigación.

- **Datos de estado:** Datos que definen un estado interno de la unidad de prueba y se usan para establecer ese estado o compararlo con estados existentes.
- **Objetivo de la prueba:** Un conjunto identificado de características de software que se medirán bajo condiciones específicas al comparar el comportamiento real con el comportamiento requerido descrito en la documentación del software.
- **Arquitectura de conjunto de pruebas:** Las relaciones anidadas entre conjuntos de casos de prueba que reflejan directamente la descomposición jerárquica de los objetivos de prueba.
- **Documentación de requisitos de la unidad:** documentación que establece la funcionalidad, la interfaz, el rendimiento, y requisitos de restricción de diseño para la unidad de prueba.

3. Actividades de las pruebas de unidad

Actividades involucradas en el proceso de las pruebas de software

1) Realizar la fase de planificación de la prueba
    a) Planea el enfoque general, fuentes y calendarios
    b) Determinar las características a ser probadas
    c) Refinar el plan general
2) Hacer la fase del conjunto de pruebas
    a) Diseñar el conjunto de pruebas
    b) Implementar el plan y diseño refinado
3) Medir la fase de pruebas de unidad
    a) Ejecutar los procedimientos de prueba
    b) Verificar la terminación
    c) Evaluar el esfuerzo de la prueba y unidad

ANTES DE EMPEZAR CON LAS TAREAS...

- Cuando se va a probar más de una unidad , la actividad del Plan debe abordar el conjunto total de unidades de prueba y no debe repetirse para cada unidad de prueba.
- Las otras actividades se deben realizar al menos una vez por cada unidad.
- Al realizar cualquiera de las actividades de manera inadecuada, puede llevar a rehacer una o más actividades anteriores y luego regresar a la actual. No aplica para la de planificación
- Durante el proceso de las pruebas, se debe desarrollar una especificación de diseño de prueba y un informe de resumen de prueba.

- Se pueden desarrollar otros documentos de prueba.
- Todos los documentos de prueba deben cumplir con ANSI / IEEE Std 829-1983.
- todos los documentos de prueba deben tener autores identificados y estar fechados.
- La especificación del diseño de prueba derivará su información de las actividades Determinar, Refinar y Diseñar.
- El informe de resumen de prueba obtendrá su información de todas las actividades

3.1 PLANEAR EL ENFOQUE GENERAL, RECURSOS Y CALENDARIOS

- La planificación de la prueba de la unidad general debe realizarse durante la planificación general de la prueba y debe registrarse en el documento de planificación correspondiente.

ENTRADAS

- Planes del proyecto
- Documentación de requerimientos de software

TAREAS DEL PLAN

1. ***Especificar el enfoque general para las pruebas de unidad:***
    - Identificar las áreas de riesgo que serán abordadas por la prueba.
    - Especificar restricciones en la determinación de características, diseño de pruebas o implementación de pruebas
    - Identificar las fuentes existentes de entrada, salida y datos de estado
    - Identificar técnicas generales para la validación de datos. y para ser utilizado para el registro, recopilación y validación de resultados
2. ***Especificar requerimientos de 'completitud':***
    - Áreas a ser cubiertas por pruebas de unidad
    - Grado de cobertura requerida para cada área
    - Cada característica debe ser cubierta por caso de prueba o excepción aprobada.
3. ***Especificar requerimientos de terminación:***
    - Para terminación 'normal' del proceso
    - Condiciones que pueden causar terminación anormal y procedimientos a aplicar.

4. ***Determinar requerimientos de recursos:***
    - Estimar recursos requeridos para adquisición de pruebas, ejecución inicial y repetición de las actividades de pruebas.
    - Hardware, herramientas de prueba, archivos, formatos, etc.
    - Recursos que necesitan preparación y responsables
    - Arreglos para los recursos, incluyendo peticiones

5. ***Especifica agenda general:***
    - Restringida por recursos y la disponibilidad de tests de unidad.

Salidas del plan

- Generar info de planeación de test de unidad de los puntos anteriores 1-5
- Peticiones de recursos si generados en punto 4 anterior (opcional)

3.2 DETERMINAR CARACTERÍSTICAS A SER PROBADAS

ENTRADAS

- Documentación de requerimientos de unidad
- Documentación de diseño de la arquitectura de software (opcional)

TAREAS

1. ***Estudiar requerimientos funcionales:*** cada función descrita en documentación de requerimientos, cada una con ID único.
2. ***Identificar requerimientos adicionales y procedimientos asociados:*** otros requerimientos como atributos de calidad y restricciones de diseño, asociados a características de software.
3. ***Identificar estados de la unidad:*** software inactivo, listo para recibir, procesando, y sus transiciones.
4. ***Identificar datos de entrada y salida, y sus características***: rangos válidos, formatos, relaciones entre campos.
5. ***Seleccionar elementos a ser incluidos en las pruebas:*** incluye datos de entrada válidos e inválidos. Identificar riesgos de elementos no seleccionados.

SALIDAS

- Lista de requerimientos incluidos en las pruebas
- Aclaraciones de requerimientos (opcional)

3.3 REFINAR/DETALLAR EL PLAN GENERAL

ENTRADAS

- Lista de elementos incluidos
- Información general de la planeación de tests de unidad

TAREAS

1. ***Refinar enfoque:*** Identificar casos de prueba existentes y procedimientos de test para ser considerados para su uso. Identificar técnicas para validación de datos, registro de salidas, etc.
2. ***Especificar requerimientos de recursos especiales:*** Identificar recursos necesitados para tests de unidad.
3. ***Especificar agenda detallada:*** Basada en software de apoyo, recursos especiales, disponibilidad de unidad y agenda de integración.

SALIDAS

- Especificar información de tests de unidad (puntos 1 a 3 anteriores)
- Peticiones de recursos especiales

3.4 DISEÑAR EL CONJUNTO DE PRUEBAS

ENTRADAS

- Documentación de requerimientos
- Lista de elementos ser incluidos en las pruebas
- Información de planeación de tests
- Documentación de diseño de unidad
- Especificaciones de pruebas previamente aplicadas (si están disponibles)

TAREAS DE DISEÑO

1. ***Diseñar arquitectura del conjunto de pruebas:*** Basándose en características a probar y condiciones especificadas por los elementos elegidos. Diseñar objetivos para que cada objetivo de bajo nivel pueda ser probado por algunos casos de prueba. Registrar jerarquía de objetivos.
2. ***Obtener procedimientos de pruebas explícitos como se requiera:*** Documentación de requerimientos, información del plan de pruebas y especificaciones de casos de prueba.
3. ***Obtener especificaciones de casos de prueba:*** Especificar nuevos casos de prueba.
4. ***Aumentar el conjunto de especificaciones de casos de prueba basado en información de diseño:*** basándose en el diseño de la unidad, actualizar el conjunto de pruebas de acuerdo al punto (1).
5. ***Completar la especificación del diseño de pruebas:*** completarlo de acuerdo al ANSI/IEEE 829-1983 (Estándar para la documentación de pruebas de software)

SALIDAS

- Especificación de diseño de pruebas de unidad
- Especificación separada de los procedimientos de pruebas
- Especificación de casos de prueba
- Peticiones de mejora de diseño de unidad

3.5 IMPLEMENTAR EL PLAN REFINADO Y EL DISEÑO

ENTRADAS

- Información de planeación de pruebas de unidad
- Especificaciones de los casos de prueba
- Descripciones de estructura de datos de software
- Recursos de apoyo para pruebas
- Ítems de prueba
- Datos y herramientas de prueba de actividades de prueba anteriores (si disponibles)

TAREAS DE IMPLEMENTACIÓN

1. ***Obtener y verificar datos de testeo:*** Generar nuevos datos requeridos. Verificar consistencia e integración de datos contra su especificación.
2. ***Obtener recursos especiales:*** Obtener recursos de test especificados en 3.3
3. ***Obtener items de prueba:*** Coleccionar ítems de prueba ej. manuales de usuario, procedimientos del SO, software que interfiere con los tests de unidad.

SALIDAS

- Datos de pruebas verificados
- Recursos de apoyo de pruebas
- Configuración de ítems de prueba
- Resumen de los ítems

3.6 EJECUTAR LOS PROCESOS DE PRUEBAS

ENTRADAS

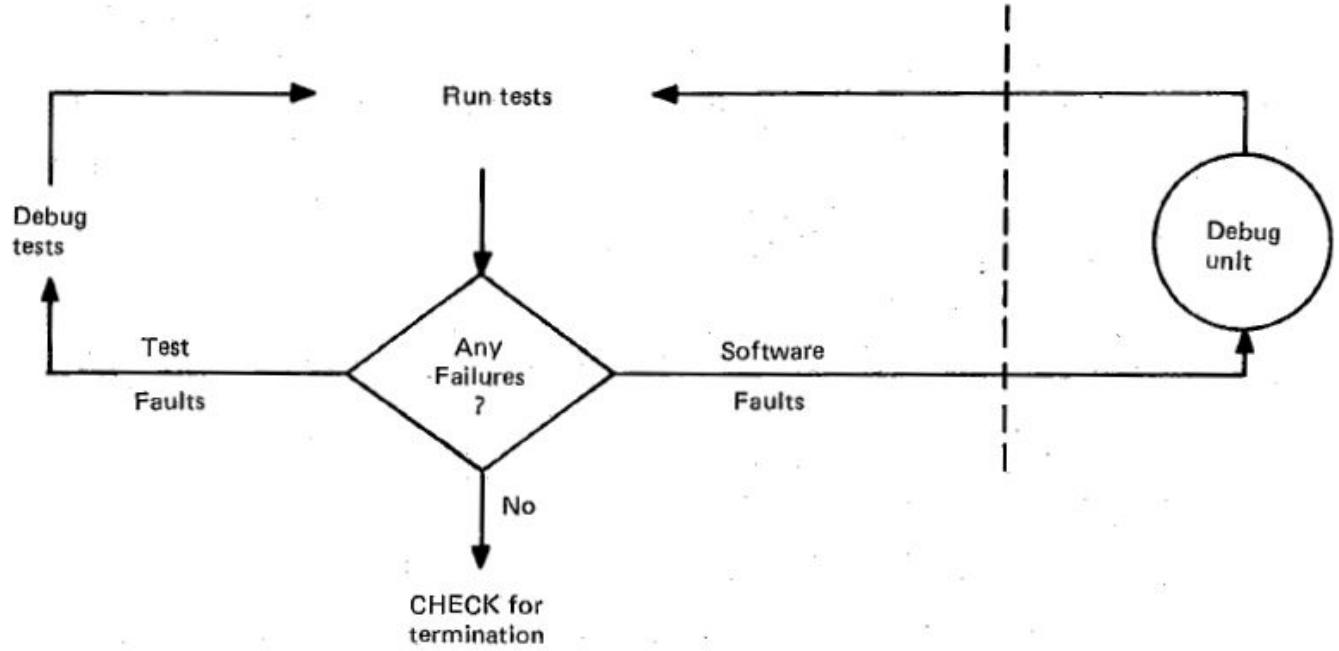
- Datos de pruebas verificados
- Recursos de apoyo de pruebas
- Configuración de ítems de prueba
- Especificaciones de casos de pruebas

TAREAS DE IMPLEMENTACIÓN

1. ***Ejecutar test:*** Preparar ambiente de pruebas. Correr el conjunto de pruebas. Registrar todos los incidentes en el *Resumen de resultados*.
2. ***Determinar resultados:*** Para cada caso de prueba, determinar si la unidad pasó o no basándose en las especificación de los casos de prueba.

   Cada fallo puede estar asociado a uno de los siguientes casos:
   1. Fallo de especificación del caso de prueba o especificación de datos.
   2. Fallo en la ejecución del procedimiento de la prueba
   3. Fallo en el ambiente de prueba
   4. Fallo en la implementación de la unidad
   5. Fallo en la unidad de diseño

Determinar los resultados:



SALIDAS

- Información de la ejecución: salidas de las pruebas, resultados del análisis de fallas, actividades de corrección, razones de fallas no corregidas, etc.
- Especificaciones de pruebas revisadas
- Datos de prueba revisados

3.7 COMPROBAR TERMINACIÓN

ENTRADAS

- Completitud y terminación de requerimientos
- Información de ejecución
- Especificaciones de las pruebas
- Especificaciones de estructura de datos de software

TAREAS DE IMPLEMENTACIÓN

1. ***Checar terminación normal del proceso de pruebas:*** Determinar si se necesitan más pruebas checando requerimientos de completitud o preocupaciones de historial de fallas.
2. ***Checar terminación anormal del proceso de pruebas:*** Especificar porque se dio la terminación anormal junto con pruebas no terminadas y fallas no corregidas.
3. ***Complementar el conjunto de pruebas:*** Cuando se necesitan pruebas adicionales.

SALIDAS

- Información del reporte de resumen de pruebas, debe incluir condiciones de terminación y cualquier actividad para añadir pruebas.
- Especificaciones de pruebas adicionales
- Datos de prueba adicionales

3.8 EVALUAR EL ESFUERZO DE PRUEBA Y UNIDAD

- ***Evaluar entradas***

1. Especificación de diseño de prueba unitaria
2. Ejecución de información
3. Información de verificación
4. Especificaciones de casos prueba separadas

- ***Evaluar tareas***

1. **Describir el estado de prueba:** Registrar las variantes de los planes de prueba y especificaciones de prueba en la sección "variaciones del resumen del reporte de pruebas ” especificando la razón de cada varianza
2. **Describir el estado de la unidad:** Registrar las diferencias entre el resultado de pruebas y el documento.
3. **Completar el resumen del reporte de prueba:** Complete el informe de resumen de prueba para la unidad de acuerdo con ANSI / IEEE Std 829-1983 [2].
4. **Asegurar la preservación de los productos de prueba:** Asegúrese de que los productos de prueba se recopilan, organizan y almacenan para referencia y reutilización.

- ***Evaluar salidas***

1. Completar el resumen del reporte de prueba
2. Colección completa y almacenada de productos de prueba

Apéndice a: Pautas de implementación y uso

- ***A1. Uso del estándar***

1. Como base de comparación para confirmar las prácticas actuales
2. Como fuente de ideas para modificar las prácticas actuales
3. Como reemplazo de las prácticas actuales

- ***A2. Requerimientos de prueba adicionales***

➔ Los requisitos como la cantidad de documentación de prueba adicional, el nivel de detalle que debe incluirse, el número, tipo de aprobaciones y revisiones deben especificarse para cada proyecto.

➔ Si los requisitos son específicos del proyecto, deben aparecer en el plan del proyecto, el plan de aseguramiento de la calidad, el plan de verificación y validación o el plan de prueba general.

- ***A3. Documentación adicional de pruebas***

➔ La información contenida en la especificación de diseño de prueba y el informe de resumen de prueba se considera un mínimo absoluto para la visibilidad del proceso.

➔ Se asume que cualquier necesidad de información de prueba puede ser satisfecha por el conjunto de documentos de prueba especificados en ANSI / IEEE Std 829-1983.

- ***A4. Aprobaciones y revisiones***

Si se desea más control, se deben considerar las siguientes tareas adicionales:

1. Aprobación del enfoque general al final del plan
2. Aprobación de los requisitos identificados al final de Determinar la aprobación de planes específicos al final de Refinar
3. Aprobación de las especificaciones de prueba al final del diseño
4. Revisión de la preparación para la prueba al final de la implementación
5. Revisión del informe resumido de la prueba al final de la evaluación

- ***A5. Vigilar auditoría***

➔ El conjunto de documentos de prueba generados junto con los informes de las revisiones de prueba debe ser suficiente para proporcionar toda la información de auditoría requerida.

- ***A6. Gestión de configuración***

➔ La administración de la configuración debe ser la fuente de los requisitos del software, el diseño arquitectónico del software, la estructura de datos del software y la documentación de requisitos de la unidad.

➔ Los productos de prueba final de la unidad se deben proporcionar a la gestión de la configuración.

- ***A7. Determinación de las características basadas en los requisitos***

➔ Factores psicológicos pueden hacer que sea muy difícil para el desarrollador de la unidad determinar un conjunto efectivo de elementos basados en requisitos para ser incluidos en las pruebas. Hay varias formas de organizar esta separación:

1. Los desarrolladores determinan estos elementos el uno para el otro.
2. Los desarrolladores prueban completamente el código de cada uno.
3. Debe haber disponible un grupo de prueba por separado. **El tamaño del proyecto o la importancia del software pueden determinar si un grupo separado puede estar justificado**
4. Si los desarrolladores determinan los elementos basados en los requisitos para su propio software, deben realizar esta determinación antes de que comience el diseño del software.

- ***A8. Involucramiento del usuario***

Si la unidad que se probará interactúa con los usuarios, puede:

➔ Ser muy eficaz involucrar a los usuarios para determinar los elementos basados en los requisitos que se incluirán en la prueba.

- ***A9. Requisitos de cobertura basados en códigos más estrictos***

➔ Según lo crítica que es la unidad o la escasez de requisitos de esta y la información de diseño se podría reforzar el requisito de cobertura basado en código.

- ***A10. Herramientas de cobertura de código***

➔ Se recomienda encarecidamente un medio automatizado para registrar la cobertura del código fuente durante la ejecución de la prueba unitaria.

➔ La automatización suele ser necesaria porque el análisis de cobertura manual no es confiable y no es económico.

➔ El informe identifica las instrucciones no ejecutadas.

- ***A11. La mejora de procesos***

Para evaluar y mejorar la efectividad de las pruebas unitarias, se recomienda:

➔ Recopilar datos de fallas de los procesos que siguen a las pruebas unitarias, como la prueba de integración, la prueba del sistema y el uso de la producción.

➔ Esta información debe analizarse para determinar la naturaleza de las fallas que deberían haberse detectado mediante pruebas unitarias pero que no lo fueron.

- ***A12. Adoptando el Estándar***

➔ Para implementar con éxito un proceso de prueba basado en este estándar, se debe desarrollar una estrategia de implementación y adaptar el estándar.

➔ Ambas actividades deben reflejar la cultura y las habilidades actuales de la organización.

➔ El éxito a largo plazo requerirá un compromiso de la administración, políticas de apoyo, herramientas, capacitación y consultoría de puesta en marcha.

- ***A13. Sentido práctico del estándar***

Esta norma representa el consenso sobre la definición de buena práctica de ingeniería de software.

➔ Comenzar a utilizar el estándar implica nuevas políticas, nuevos estándares y procedimientos, nuevas herramientas y nuevos programas de capacitación.

➔ Si las diferencias entre la práctica estándar y la actual son demasiado grandes, entonces los cambios tendrán que introducirse gradualmente.

➔ La respuesta a la cuestión del sentido práctico del estándar es básicamente un nivel de deseo.. ¿Qué tanto una organización desea obtener el control de su unidad de prueba?.

Apéndice b: Conceptos y suposiciones

B1. Conceptos de ingeniería de software

- ***B1.1 Relación de las pruebas con la verificación y validación***

Las pruebas son solo una de varias actividades complementarias de verificación y validación.
Otras actividades incluyen :

➔ Revisiones técnicas (por ejemplo, inspecciones de código)
➔ Análisis estático y prueba de corrección.

La especificación de un proceso integral de verificación y validación está fuera del alcance de esta norma.

- ***B1.2 Pruebas como desarrollo de productos***

Las pruebas incluyen un proceso de desarrollo de productos.

➔ Resulta en un conjunto de prueba compuesto de datos, software de soporte de prueba y procedimientos para su uso.
➔ Este producto está documentado por especificaciones e informes de prueba. Al igual que con cualquier proceso de desarrollo de productos, el desarrollo del conjunto de pruebas requiere planificación, requisitos, diseño, implementación y evaluación.

- ***B1.3 Composición de la depuración***

El proceso de depuración se compone de dos actividades principales:

➔ El análisis de fallas: localizar e identificar todas los errores responsables de esa falla

➔ Eliminar todas las fallas identificadas y evitar la introducción de nuevas.

La especificación del proceso de análisis de fallas o corrección de fallas está fuera del alcance de esta norma.

- ***B1.4 Relación de las pruebas con la depuración***

➔ Las pruebas pueden necesitar los resultados del análisis de fallas de la depuración para decidir un curso de acción.

➔ Esas acciones pueden incluir la finalización de las pruebas o una solicitud de cambios de requisitos o corrección de fallas.

- ***B1.5 Relación entre tipos de unidades***

➔ No es necesaria una relación uno a uno entre unidades de diseño, unidades de implementación y unidades de prueba.

➔ Varias unidades de diseño pueden constituir una unidad de implementación (por ejemplo, un programa) y varias unidades de implementación pueden constituir una unidad de prueba.

- ***B1.6 Necesidad de información de diseño e implementación***

A menudo, la información de los requisitos no es suficiente para realizar pruebas efectivas, aunque, fundamentalmente, las pruebas miden el comportamiento real frente al comportamiento requerido.
Con frecuencia, se necesita información sobre:

➔ El diseño
➔ La implementación

- ***B1.7 Especificación incremental de los elementos a considerar en las pruebas***

Los elementos a considerar en las pruebas se pueden acumular de manera incremental durante diferentes períodos de actividad de prueba. Para las implementaciones de lenguaje de procedimientos (por ejemplo, COBOL), la especificación del elemento se produce en tres incrementos:

1. El primer grupo se especifica durante la actividad de “Determinar” y se basa en la documentación de requisitos de la unidad.
2. El segundo grupo se especifica durante la actividad de “Diseño” y se basa en el diseño de la unidad (es decir, algoritmos y estructuras de datos) .
3. El tercer grupo se especifica durante la actividad “Verificar” y se basa en el código de la unidad.

- ***B1.7 Especificación incremental de los elementos a considerar en las pruebas***

Para las implementaciones de lenguaje no procedimental (por ejemplo, escritor de informes u otros lenguajes de especificación), la especificación se produce en dos incrementos:

1. Durante la actividad Determinar y se basa en los requisitos y
2. Durante el Diseño y se basa en la especificación no procedimental.

- ***B1.8 Creación incremental de una especificación de diseño de prueba***

➔ La información registrada en la especificación de diseño de prueba se genera durante las actividades Determinar, Refinar y Diseñar.

➔ A medida que progresa cada una de estas actividades de prueba, la información se registra en las secciones apropiadas de la especificación.

➔ El documento completo debe estar completo al final de la iteración final de la actividad de Diseño.

- ***B1.9 Creación incremental del informe de resumen de prueba***

➔ La información registrada en el informe de resumen de pruebas se genera durante todas las actividades de prueba de la unidad.

➔ El informe se inicia durante la implementación, se actualiza durante la ejecución y el chequeo, posteriormente se completa durante la evaluación.

B2. Probando suposiciones

El enfoque de pruebas unitarias especificado en este estándar se basa en una variedad de suposiciones económicas, psicológicas y técnicas.

- ***B2.1 El objetivo de las pruebas unitarias es intentar determinar la corrección e integridad de una implementación con respecto a los requisitos de la unidad y la documentación de diseño tratando de descubrir fallas en:***
  1. Las características requeridas de la unidad en combinación con sus estados asociados (por ejemplo, inactivo, activo esperando un mensaje, procesamiento activo de un mensaje.
  2. El manejo de la unidad de entradas inválidas
  3. Cualquier uso o procedimiento operativo asociado solo con la unidad
  4. Los algoritmos de la unidad o las estructuras internas de datos, o ambos
  5. Los límites de decisión de la lógica de control de la unidad

- ***B2.2 Pruebas***

➔ Medir el comportamiento real asociado con una interfaz, estado o requisito, contra el comportamiento requerido correspondiente.

➔ Cualquier proceso de prueba de unidad verificable debe tener requisitos documentados para la unidad de prueba.

Este estándar asume que la documentación de los requisitos de la unidad existe antes de que comience la prueba.

- ***B2.3***

➔ La documentación de los requisitos de la unidad debe revisarse minuciosamente para verificar que esté completa, comprobable y rastreable.

➔ Esta norma supone que los requisitos han sido revisados como parte normal del proceso de revisión de la documentación o en una revisión de requisitos de unidad especial.

- ***B2.4***

Existen importantes beneficios económicos en la detección temprana de fallas. Esto implica que:

➔ El desarrollo del conjunto de pruebas debe comenzar tan pronto como sea posible después de la disponibilidad de la documentación.

- ***B2.5***

Los niveles de prueba del proyecto (por ejemplo, aceptación, sistema, integración, unidad) se especifican en:

1. Planes del proyecto
2. Planes de verificación y validación
3. Planes de prueba generales.

- ***B2.6***

➔ La disponibilidad de insumos y recursos para realizar una tarea es la principal limitación en la secuencia de actividades y en la secuencia de tareas dentro de una actividad.

➔ Si los recursos necesarios están disponibles, algunas de las actividades y tareas dentro de una actividad pueden realizarse simultáneamente.

- ***B2.7***

Esta norma asume que, en general:

➔ Es más rentable reducir el diseño de los casos de prueba en función de las características del código fuente hasta que se haya ejecutado el conjunto de casos de prueba basados en los requisitos y las características del diseño.

➔ Este enfoque minimiza la tarea de diseño basada en código.

Apéndice C: técnicas y herramientas

Cl. general

Las herramientas de software { Programas para computadora }

Las técnicas y herramientas de software se pueden usar y reutilizar en una variedad de entornos de desarrollo. Su uso efectivo aumenta la productividad de la ingeniería y la calidad del software.

Técnicas de software { Métodos detallados que ayudan en la especificación, construcción, prueba, análisis, administración, documentación y mantenimiento de otros programas de computadora }

PREGUNTAS

- ¿En qué consiste el objetivo principal del estándar?
- ¿Cuáles son las 3 fases en las que se divide el estándar?
- ¿Cual es el único estándar al que hace referencia?