

Leyenda

Primera columna	Técnicas de Verificación
Segunda y tercera columna	Técnicas de Validación

Walkthrough

Revisión informal de documentos o código con el equipo

Fases En todo el CVDS

Fortalezas Retroalimentación rápida, útil para compartir conocimiento

Debilidades Poco formal y depende de la experiencia del equipo

Code inspections

Detectar errores y mejorar la calidad mediante revisiones detalladas del código

Fases Codificación

Fortalezas Alta precisión para encontrar defectos

Debilidades Requiere tiempo extra y planificación

Reviews

Inspección sistemática del software

Fases Primeras etapas del CVDS

Fortalezas Reduce el costo de mantenimiento del software

Debilidades Puede ser costosa y depende de la disponibilidad del equipo

Formal Proofs

Justificar que un programa cumpla una especificación formal de su comportamiento

Fases Diseño y codificación

Fortalezas Mejor documentación y comprensión

Debilidades No siempre viables para todos los proyectos

Simulation and prototyping

Es una técnica que consiste en construir versiones para analizar diseños de producto candidatos

Fases Diseño

Fortalezas Feedback temprano de las opciones de diseño

Debilidades Puede requerirse mucho recursos para analizar cada opción

Black Box Testing

Evaluar la funcionalidad sin ver el código

Fases Fase de pruebas funcionales

Fortalezas No requiere conocimiento interno del sistema

Debilidades No detecta problemas estructurales

White Box Testing

Evaluar la estructura interna del código

Fases Fase de pruebas de unidad y de integración

Fortalezas Permite identificar errores internos y optimizar el código

Debilidades Requiere acceso al código y conocimientos técnicos avanzados

Heuristic Testing

Detectar fallos usando la experiencia y el conocimiento del sistema

Fases Fase de pruebas de sistema

Fortalezas Útil para identificar problemas difíciles de prever

Debilidades Depende de la habilidad del probador

Interface Testing

Validar la interacción entre módulos o sistemas

Fases Fase de pruebas de integración

Fortalezas Asegura la correcta interacción de componentes

Debilidades Puede ser difícil simular todos los escenarios de interfaz

Equivalence Class Partitioning

Divide los datos de entrada en clases que se consideran equivalentes

Fases Pruebas de sistema

Fortalezas Reduce casos de prueba

Debilidades Omite errores fuera de clases

Boundary Value Analysis

Prueba los valores en los límites de los rangos de entrada

Fases Pruebas de sistema

Fortalezas Detecta errores en límites de entrada

Debilidades No cubre casos intermedios

Decision Table-Based Testing

Usa tablas para representar combinaciones de condiciones y acciones

Fases Diseño de pruebas, pruebas de requisitos

Fortalezas Maneja lógica compleja

Debili- Tablas grandes difíciles de
dades gestionar

Cause Effect Graphing

Crea un grafo para mostrar relaciones lógicas entre condiciones y efectos

Fases Análisis y diseño de pruebas

Fortalezas Visualiza condiciones y efectos

Debili- Difícil con requisitos
dades ambiguos

DD Path Testing

Usado para diseñar casos de prueba basados en rutas de ejecución independientes (flujo del programa)

Fases Requerimientos y diseño

Fortalezas Proporciona una cobertura completa de las ramas

Debili- No logra cubrir todas las rutas
dades posibles del grafo del flujo de control

Data Flow Testing

Analiza como las variables son definidas y usadas a lo largo del funcionamiento de un programa.

Fases Pruebas

Fortalezas Detección precisa de errores lógicos

Debili- Limitada debido a que no
dades detecta todo tipo de errores

References

G. R. Maquieira, "Qué es black box testing o pruebas de caja negra", Openwebinars.net, 06-ene-2023.

Edu.uy. [En línea]. Disponible en: <https://www.fing.edu.uy/tecnoinf/maldonado/cursos/ingsoft/materiales/teorico/is09-Verificacion-Validacion.pdf>. [Consultado: 06-nov-2024].

G. Sharma, "Data Flow Testing," GeeksforGeeks, Aug. 14, 2023. [Online]. Available: <https://www.geeksforgeeks.org/data-flow-testing/>

S. Sharma, "Path Testing in Software Engineering," GeeksforGeeks, Jul. 20, 2023. [Online]. Available: <https://www.geeksforgeeks.org/path-testing-in-software-engineering/>

TryQA, "What is Verification in Software Testing (or) What is Software Verification?," TryQA.com. [Online]. Available: <https://tryqa.com/what-is-verification-in-software-testing-or-what-is-software-verification/>

ChipVerify, "Verification Techniques," ChipVerify.com. [Online]. Available: <https://www.chipverify.com/verification/verification-techniques>

S. K. Chopra, Software Quality Assurance: Principles and Practice, 1st ed. New Delhi, India: Katson Publishing House, 2018.

Not published yet.

Last updated 6th May, 2025.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>