

Fundamentos de Testing

Instructora: Ing. Gabriela Bagatello



Agenda

Unidad 3: Técnicas estáticas

3.1 Técnicas estáticas y el proceso de pruebas

3.2 Proceso de revisión

3.3 Análisis estático con herramientas



Temas: 3.1 Técnicas estáticas y el proceso de pruebas

Las pruebas estáticas:

- ❖ Se basan en el examen manual (revisiones)
- ❖ Y en el análisis automatizado (análisis estático) del código o cualquier otra documentación del proyecto sin ejecutar el código
- ❖ Los defectos detectados durante las revisiones al principio del ciclo de vida a menudo son más baratos de eliminar que los detectados durante las pruebas realizadas ejecutando el código



Temas: 3.1 Técnicas estáticas y el proceso de pruebas

Productos de trabajo de software que pueden ser objeto de una revisión:

- Especificaciones de requisitos
- Especificaciones de diseño
- El código
- Planes de prueba
- Especificaciones de prueba
- Casos de Prueba
- Guías de usuario
- Páginas Web



Temas: 3.1 Técnicas estáticas y el proceso de pruebas

Beneficios de las revisiones:

- ✓ Detección y corrección temprana de defectos
- ✓ Desarrollo de mejoras de productividad
- ✓ Reducción de los tiempos de desarrollo
- ✓ Ahorro de tiempo y dinero invertido en pruebas
- ✓ Menos defectos
- ✓ Comunicación mejorada



Temas: 3.1 Técnicas estáticas y el proceso de pruebas

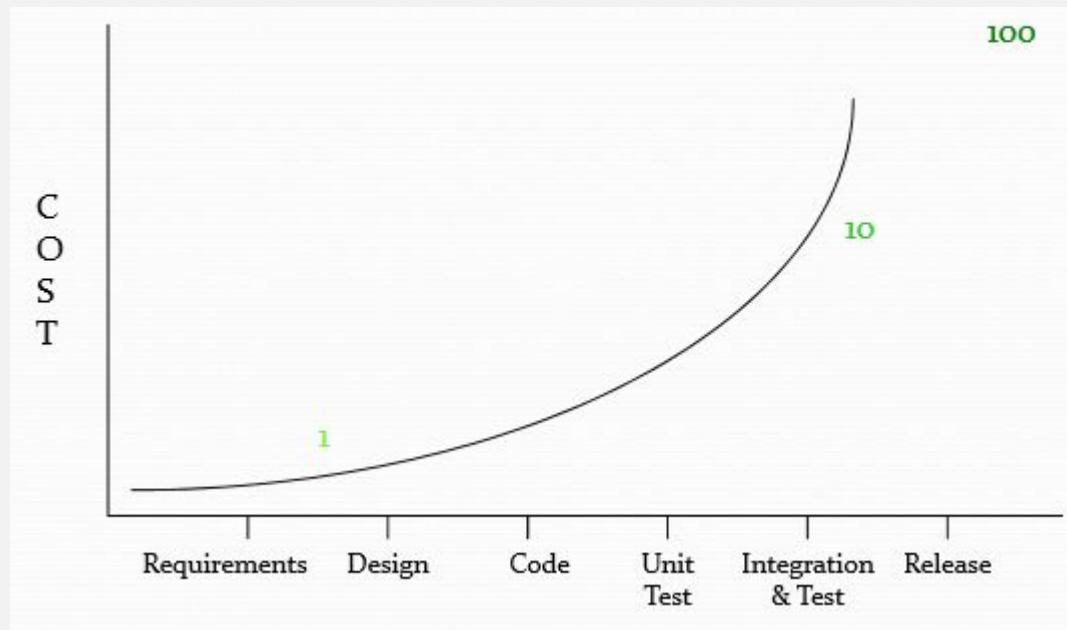
Defectos típicos fáciles de localizar en revisiones:

- Desviaciones de los estándares
- Defectos de requisito
- Defectos de diseño
- Mantenibilidad insuficiente
- Especificaciones de interfaz incorrectas



Temas: 3.1 Técnicas estáticas y el proceso de pruebas

Relación de Costo de arreglar un Falla:



Temas: 3.1 Técnicas estáticas y el proceso de pruebas

Ventajas:

- Costos más bajos y un potencial de ahorro relativamente alto
- Defectos en la documentación son detectados y corregidos en forma temprana
- Los documentos de alta calidad mejoran el proceso de desarrollo



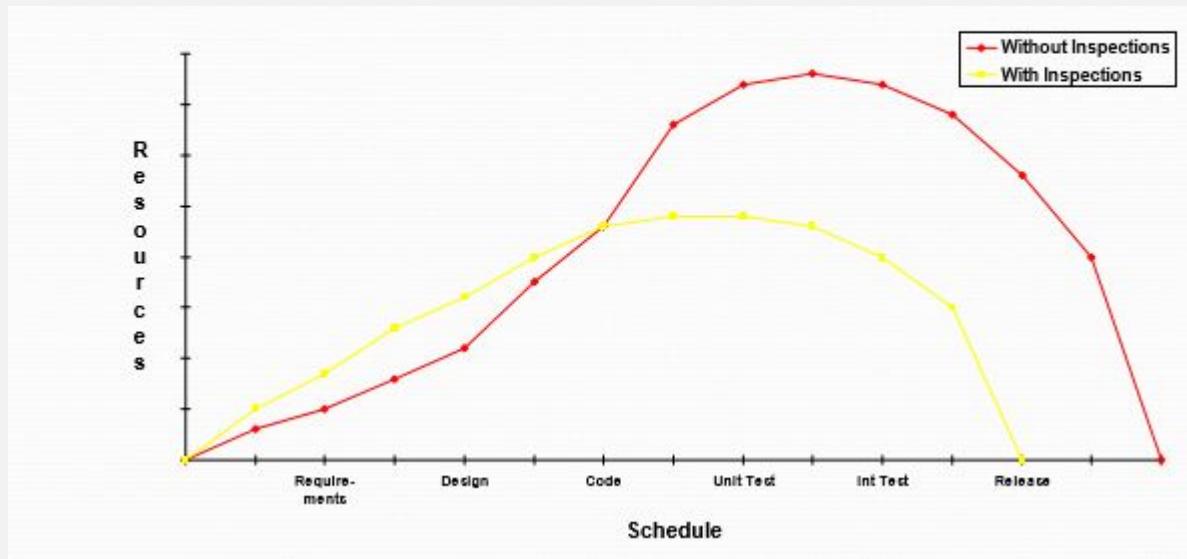
Temas: 3.1 Técnicas estáticas y el proceso de pruebas

Desventajas:

- Se podrían presentar situaciones de tensión en el caso de enfrentamientos directos con el autor
- Los expertos involucrados en las revisiones deben adquirir conocimientos específicos del producto
- Inversión considerable de tiempo
- Moderador y participantes influyen directamente en la calidad de la revisión

Temas: 3.1 Técnicas estáticas y el proceso de pruebas

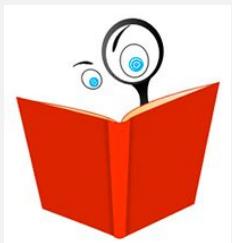
Revisiones – Rentabilidad de su uso



Temas: 3.2 Proceso de revisión

3.2.1 Fases de una revisión

- ❖ Fase de planificación (“Planning”)
- ❖ Preparación de la organización e inicio (“Kick-off”)
- ❖ Preparación individual (“Preparation”)
- ❖ Reunión de revisión (“Review meeting”)
- ❖ Reconstrucción – Repetición del trabajo (“Rework”)
- ❖ Seguimiento (“Follow up”)



Temas: 3.2 Proceso de revisión

Actividades de una revisión formal

1. Planificar

- Definir los criterios de revisión
- Seleccionar al personal
- Asignar funciones

2. Definir los criterios de entrada y salida para mas tipos de revisión formal

- Seleccionar que partes de los documentos deben revisarse

3. Inicio

- Repartir los documentos
- Explicar los objetivos, procesos y documentos a los participantes

Temas: 3.2 Proceso de revisión

4. Comprobar los criterios de entrada
5. Preparación individual
 - Prepararse para la reunión de revisión repasando los documentos
6. Prestar especial atención a posibles defectos, preguntas y comentarios
7. Examen/evaluación/registro de los resultados
 - Debatir o registrar, mediante los documentos o actas
 - Prestar especial atención a los defectos

Temas: 3.2 Proceso de revisión

8. Examinar/evaluar y registrar durante reuniones físicas o de seguimiento de los grupos comunicaciones electrónicas
9. Adaptar
10. Corregir los defectos detectados
 - Registrar el estado actualizado de los defectos
11. Hacer un seguimiento
 - Comprobar que los defectos han sido tratados
 - Recopilar métricas
12. Comprobar los criterios de salida



Temas: 3.2 Proceso de revisión

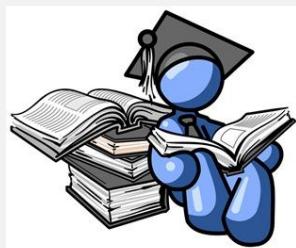
3.2.2 Funciones y responsabilidades

- ❖ Director (“Manager”): inicia la revisión, decide respecto a los participantes y asigna recursos
- ❖ Moderador (“Moderator”): dirige la reunión, hace de mediador, concluye resultados
- ❖ Autor (“Author”): expone su trabajo a la critica, lleva a cabo los cambios recomendados
- ❖ Revisores (“Reviewer”): detecta defectos, desviaciones, áreas problemáticas
- ❖ Escriba o registrador (“Scribe”): documenta todos los puntos identificados

Temas: 3.2 Proceso de revisión

3.2.3 Tipos de revisiones (IEEE 1028)

- ❖ Revisión informal (“Informal review”)
- ❖ Revisión guiada (“Walkthrough”)
- ❖ Revisión técnica (“Technical review”)
- ❖ Inspección (“Inspection”)



Temas: 3.2 Proceso de revisión

- Revisión informal (“Informal review”)
 - Ausencia de un proceso formal
 - Puede adoptar la forma de una revisión por parte de un líder técnico de los diseños y el código
 - Los resultados pueden estar documentados
 - Su utilidad varia en función de los revisores
 - Objetivo principal: forma barata de obtener beneficios



Temas: 3.2 Proceso de revisión

- ❑ Revisión guiada (“Walkthrough”)
 - ✓ Reunión liderada por el autor
 - ✓ Puede adoptar la forma de escenarios, simulacros o participación de grupos de pares
 - ✓ Escriba opcional (distinto del autor)
 - ✓ Puede variar en la práctica desde bastante informal hasta muy formal
 - ✓ Objetivos principales: aprender, entender, encontrar defectos



Temas: 3.2 Proceso de revisión

Revisión técnica (“Technical review”)

- Proceso documentado y definido para la detección de defectos
- Puede llevarse a cabo como una revisión entre pares sin la participación de la dirección
- Idealmente está dirigida por un moderador formado (distinto al autor)
- Preparación previa a la reunión por parte de los revisores
- Uso opcional de listas de comprobación



Temas: 3.2 Proceso de revisión

- Elaboración de un informe de revisión
- Puede variar en la práctica desde bastante informal hasta muy formal
- Objetivos principales:
 - Debatir, tomar decisiones, evaluar alternativas,
 - Encontrar defectos, resolver problemas técnicos,
 - Comprobar la conformidad con las especificaciones, los planes, la normativa y los estándares.

Temas: 3.2 Proceso de revisión

Inspección (“Inspection”)

- Dirigida por un moderador formado (distinto del autor)
- Generalmente celebrada como un examen entre pares
- Funciones definidas
- Incluye una recopilación de métricas
- Proceso formal basado en normas y listas de comprobación
- Criterios de entrada y salida especificados para la aceptación del producto de software

Temas: 3.2 Proceso de revisión

- Inspección (“Inspection”)
 - Preparación previa a la reunión
 - Informe de inspección en el que se incluye una lista de las conclusiones
 - Proceso de seguimiento formal
 - Lector opcional
 - Objetivo principal: identificar defectos



Temas: 3.2 Proceso de revisión

Factores de éxito de una revisión:

- Se deben desarrollar orientadas al logro de objetivos
- El autor del objeto revisado deberá ser motivado de una forma positiva por la revisión
- Uso sistemático de las técnicas y plantillas implantadas
- Uso de listas de comprobación mejorara la eficiencia de la revisión
- Para la ejecución apropiada de las revisiones será necesario un presupuesto

Resumen:

- ✓ En el transcurso de las pruebas estáticas no se ejecuta el objeto de prueba
- ✓ Las revisiones pueden tener lugar en fases tempranas del proceso de desarrollo, ellas complementan/extienden los métodos de pruebas dinámicos
- ✓ Fases una revisión:
 - Planificación
 - Preparación de la organización e inicio (“Kick-off”)
 - Preparación individual (“Preparation”)
 - Reunión de revisión (“Review meeting”)
 - Reconstrucción – Repetición del trabajo (“Rework”)
 - Seguimiento (“Follow up”)

Resumen:

- ✓ Roles y tareas para una revisión:
 - Director (“Manager”)
 - Moderador (“Moderator”)
 - Autor (“Author”)
 - Revisores (“Reviewer”)
 - Escriba o registrador
- ✓ Tipos de revisiones:
 - Revisión informal
 - Revisión técnica
 - Revisión guiada
 - Inspección

Temas: 3.3 Análisis estático con herramientas

Valor del análisis estático:

- ✓ La detección temprana de defectos antes de la ejecución de las pruebas
- ✓ Advertencia temprana sobre aspectos sospechosos del código
- ✓ Identificación de defectos que no se encuentran mediante pruebas dinámicas
- ✓ Detectar dependencias e inconsistencias en modelos de software
- ✓ Mantenibilidad mejorada del código y del diseño
- ✓ Prevención de defectos, si se aprende la lección en la fase de desarrollo

Temas: 3.3 Análisis estático con herramientas

Entre los defectos detectados por las herramientas de análisis estático se encuentran:

- Referenciar una variable con un valor indefinido
- Interfaces inconsistentes entre módulos y componentes
- Variables que no se utilizan o que se declaran de forma incorrecta
- Código inaccesible (muerto)
- Ausencia de lógica o lógica errónea (posibles bucles indefinidos)
- Construcciones demasiado complicadas
- Infracciones de los estándares de programación
- Vulnerabilidad de seguridad

Temas: 3.3 Análisis estático con herramientas

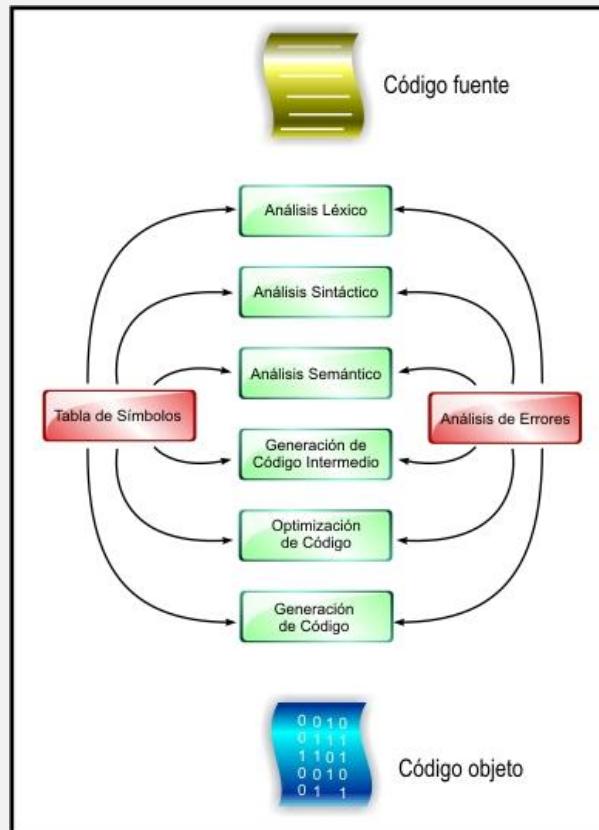
Las herramientas de análisis estático generalmente las utilizan los desarrolladores antes y durante las pruebas de componente y de integración, o durante las comprobación del código, para configurar las herramientas de gestión.

Los compiladores pueden constituir un soporte para los análisis estáticos, incluyendo el cálculo de métricas



Temas: 3.3 Análisis estático con herramientas

Compiladores:



Temas: 3.3 Análisis estático con herramientas

Análisis estático: Es aquella actividad que consiste en el análisis de un objeto de prueba (código fuente, script, requisito) llevado a cabo sin ejecutar el producto software

Possibles aspectos a ser comprobados con análisis estático:

- ❖ Reglas y estándares de programación
- ❖ Diseño de un programa (análisis de flujo de control)
- ❖ Uso de datos (análisis de flujo de datos)
- ❖ Complejidad de la estructura de un programa
- ❖ Métricas, por ejemplo numero de ciclomático



Temas: 3.3 Análisis estático con herramientas

Aspectos generales:

- ❖ Todos los objetos de prueba deben tener una estructura formal.
- ❖ El análisis estático de un programa mediante el uso de herramientas se desarrolla con un esfuerzo menor que el necesario a una inspección
- ❖ El análisis estático se ejecuta de forma previa a que tenga lugar una revisión
- ❖ Herramientas a utilizar:
 - ✓ Compiladores
 - ✓ Herramientas de análisis (Analizadores)

Temas: 3.3 Análisis estático con herramientas

→ Compiladores:

- ❖ Detecta errores sintácticos en el código fuente de un programa
- ❖ Crea datos de referencia del programa (lista de referencia cruzada, llamada jerárquica, tabla de símbolos)
- ❖ Comprueba la consistencia entre los tipos de variables
- ❖ Detecta variables no declaradas y código inaccesible

→ Analizador:

- ✓ Convenciones y estándares
- ✓ Métricas de complejidad
- ✓ Acoplamiento de objetos

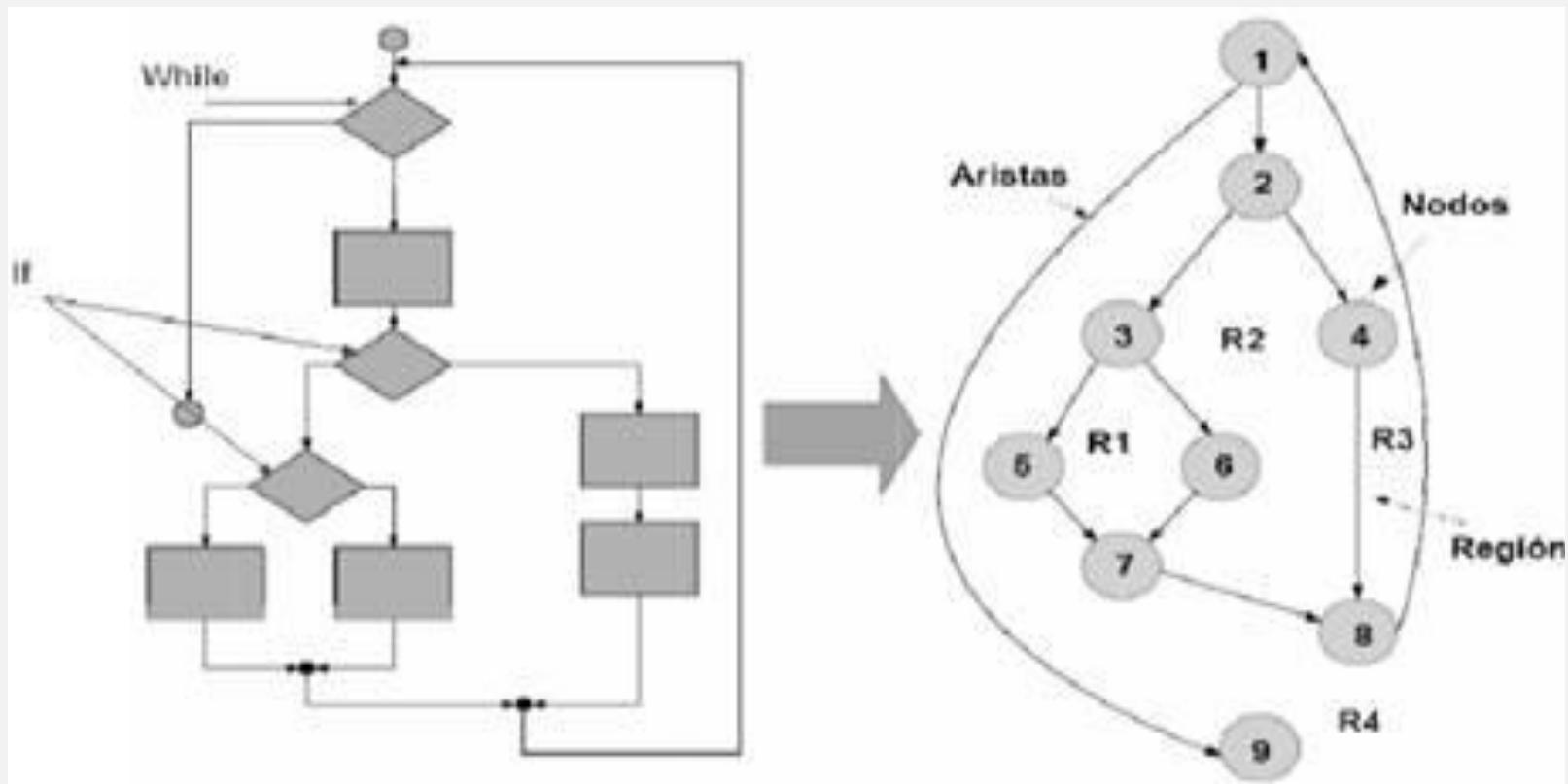
Temas: 3.3 Análisis estático con herramientas

Análisis del flujo de control:

- ❖ Propósito: detectar defectos causados por un desarrollo anómalo del código (ramas muertas, código muerto, etc)
- ❖ Método:
 - ✓ La estructura del código se representa como un diagrama de control de flujo
 - ✓ Grafo dirigido
- ❖ Resultados:
 - ✓ Visión del conjunto del código del programa comprensible
 - ✓ Las anomalías pueden ser fácilmente detectadas, los defectos se hacen evidentes.

Temas: 3.3 Análisis estático con herramientas

Análisis del flujo de control:



Temas: 3.3 Análisis estático con herramientas

Análisis del flujo de datos:

- ❖ Método:
 - ✓ Una variable x puede tener los siguientes estados a lo largo de la ejecución de un programa:
 - X se encuentra definida (d)
 - X se encuentra indefinida (u)
 - X esta referenciada (r)
 - X no ha sido utilizada (e)
 - control de flujo
 - ✓ El flujo de datos de una variable puede ser expresado como una secuencia de estados: d, u, r y e
 - ✓ Si una de estas secuencias contiene una sub-secuencia que no tiene sentido, entonces ha tenido lugar una anomalía en el flujo de datos

Temas: 3.3 Análisis estático con herramientas

Análisis del flujo de datos:

- ❖ Las métricas y su cálculo:

- ✓ Ciertos aspectos de la calidad de un programa pueden ser medidos utilizando métricas: la métrica solo tiene relevancia para el aspecto medido
- ✓ La complejidad estática de un programa puede ser medida
- ✓ Métricas diferentes tratan aspectos diferentes de la complejidad de programa

“Es difícil comparar dos métricas diferentes, incluso cuando ambas abordan el mismo atributo del programa”

Temas: 3.3 Análisis estático con herramientas

Las métricas y su implicación:

- ❖ Número ciclomático $v(G)$:

- ✓ Métrica que mide la complejidad estática de un programa basada en su grafo de flujo de control
- ✓ Mide los caminos linealmente independientes, como índice de la testabilidad y mantenibilidad
- ✓ El número ciclomático se define de la siguiente forma:

$$v(G) = e - n + 2p$$

- ✓ Valores hasta 10 son aceptables

Temas: 3.3 Análisis estático con herramientas

Ejemplo de numero ciclomático $v(G)$:

$$v(G) = e - n + 2p$$

El grafo tiene:

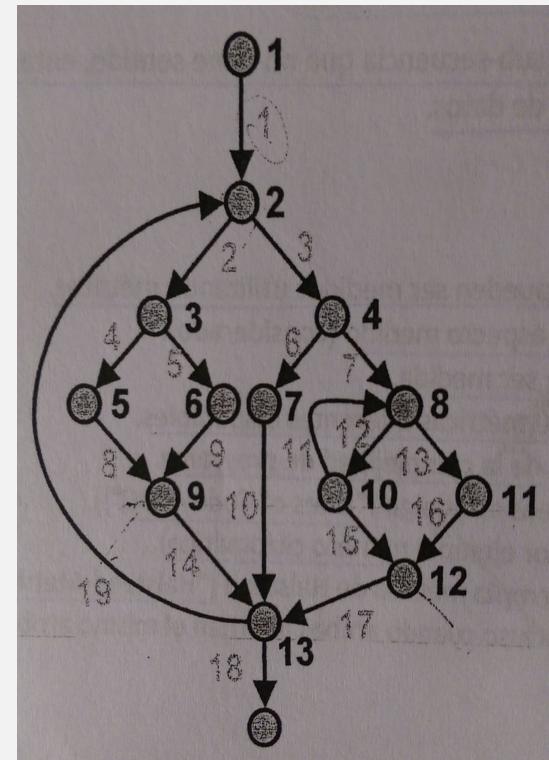
- ✓ 1 parte independiente $\rightarrow p = 1$
- ✓ 14 nodos $\rightarrow n = 14$
- ✓ 19 aristas $\rightarrow e = 19$

$$v(G) = e - n + 2p$$

$$v(G) = 19 - 14 + 2 \cdot 1$$

$$v(G) = 19 - 14 + 2$$

$$\mathbf{v(G)=7}$$



Temas: 3.3 Análisis estático con herramientas

❖ Número ciclomático $v(G)$:

- ✓ Puede ser usado como un valor objetivo para una revisión de código
- ✓ Puede ser calculado como el nro. de decisiones independientes más uno. Si las dos formas de cálculo aportan resultados diferentes se puede deber a:
 - Ramas superfluas
 - Ramas faltantes
- ✓ Aporta un índice del numero de casos de prueba necesarios (para alcanzar cobertura de decisión)

Resumen:

✓ Análisis estático:

- Con el uso de herramientas para la realización de análisis estático (compiladores, analizadores) el código del programa puede ser objeto de inspección sin ser ejecutado
- Con el uso de herramientas se puede realizar el análisis estático de un programa con un esfuerzo menor que el necesario para una inspección
- En el transcurso de las pruebas estáticas no se ejecuta el objeto de prueba

Resumen:

✓ Resultado del análisis:

- El diagrama de flujo de control presenta el flujo del programa y permite la detección de “Ramas muertas” y código inalcanzable
- Las anomalías en los datos se detectan utilizando el análisis de flujo de datos
- Las métricas pueden ser utilizadas para evaluar la complejidad estructural conduciendo a una estimación del esfuerzo en pruebas a esperar.

Agenda

Unidad 4: Técnicas de diseño de pruebas

4.1 El proceso de desarrollo de pruebas



4.2 Categorías de técnicas de diseño

4.3 Técnicas basadas en la especificación o técnicas de caja negra

4.4 Técnicas basadas en la estructura o técnicas de caja blanca

4.5 Técnicas basadas en la experiencia

4.6 Selección de la técnica de prueba

Temas: 4.1 *El proceso de desarrollo de pruebas*

Obtención de casos de prueba a partir de requisitos:

- ❖ El diseño de casos de prueba debe ser un proceso controlado
- ❖ Los casos de prueba pueden ser creados formal o informalmente, dependiendo de las características del proyecto y la madurez del proceso en uso.



Temas: 4.1 *El proceso de desarrollo de pruebas*

Trazabilidad:

- ❖ Las pruebas deben ser trazables
- ❖ Las consecuencias sobre las pruebas de los cambios a realizar en los requisitos pueden ser identificados directamente
- ❖ La trazabilidad también ayuda a determinar la cobertura de requisitos

“Los casos de prueba y los juegos de pruebas (“test suites”) son obtenidos a partir de los requisitos o características de los objetos de prueba”



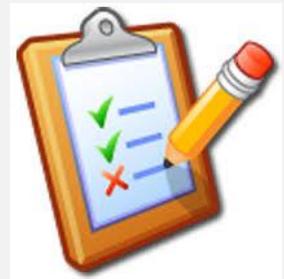
Temas: 4.1 El proceso de desarrollo de pruebas

- **Objeto de pruebas (“Test object”)**
 - elemento a ser revisado: un documento o una pieza de software en el proceso de desarrollo de software
- **Condición de prueba (“Test condition”)**
 - elemento o evento: una función, una transacción, un criterio de calidad o un elemento en el sistema
- **Criterios de prueba (“Test criteria”)**
 - el objeto de prueba debe cumplir los criterios de prueba con el objeto de superar la prueba

Temas: 4.1 El proceso de desarrollo de pruebas

Descripción de un caso de prueba según el estándar IEEE 829

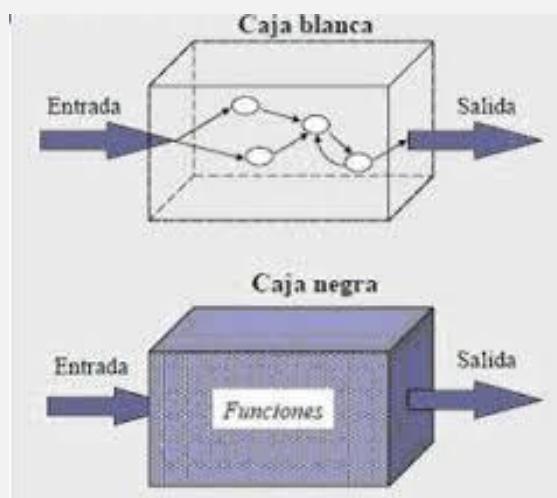
- ✓ Valores de entrada (“input values”)
- ✓ Precondiciones (“preconditions”)
- ✓ Resultados esperados (“expected results”)
- ✓ Pos condiciones (“post conditions”)
- ✓ Dependencias (“dependencies”)
- ✓ Identificador distingible (“distinct identification”)
- ✓ Requisitos (“requirements”)



Temas: 4.2 Categorías de técnicas de diseño de pruebas

- ✓ La agrupación se realiza en función del carácter básico del método utilizando para obtener los casos de prueba

- ✓ Cada grupo tiene sus propios métodos para diseñar casos de prueba



Temas: 4.2 Categorías de técnicas de diseño de pruebas

Una técnica de diseño de prueba es un procedimiento utilizado para derivar y/o seleccionar casos de prueba

- Técnicas estáticas:
 - ✓ Técnicas basadas en la estructura
- Técnicas dinámicas:
 - ✓ Técnicas basadas en la especificación
 - ✓ Técnicas basadas en la estructura
 - ✓ Técnicas basadas en la experiencia



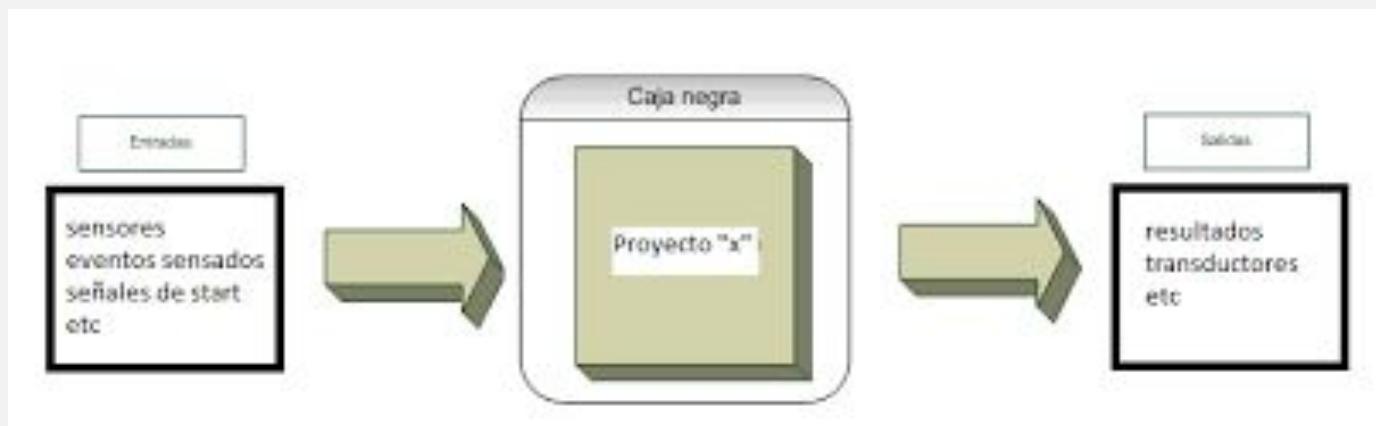
Temas: 4.2 Categorías de técnicas de diseño de pruebas

- Técnica de caja negra (“black box”)
- El probador observa el objeto de prueba como una caja negra
- Los casos de prueba se obtienen a partir del análisis de la especificación (funcional y no funcional) de un componente o sistema
- La funcionalidad es el foco de atención

Técnica de caja negra o prueba funcional o prueba
orientada a la especificación

Temas: 4.2 Categorías de técnicas de diseño de pruebas

- Técnica de caja negra (“black box”)



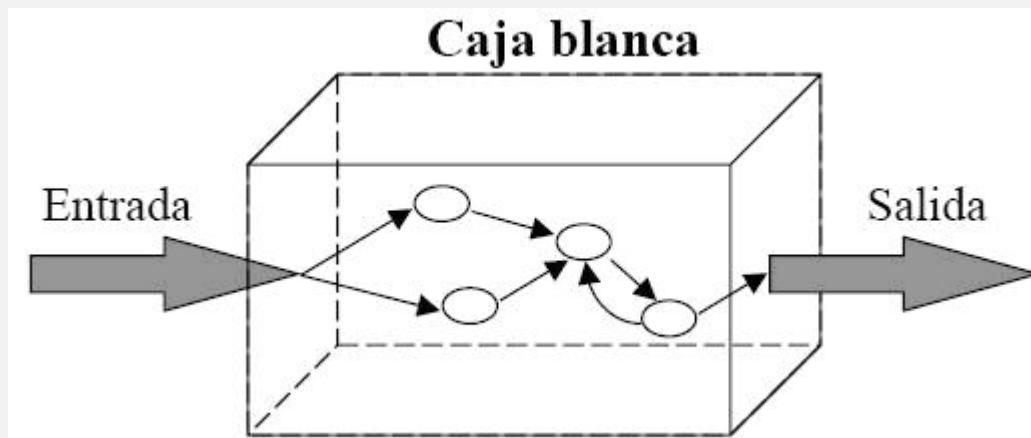
Temas: 4.2 Categorías de técnicas de diseño de pruebas

- Técnica de caja blanca (“white box”)
- El probador conoce la estructura interna del programa/código
- Los casos de prueba son seleccionados en base a la estructura interna del programa/código
- La estructura del programa es el foco de atención

Técnica de caja blanca o prueba basada en la estructura
o prueba basada en el flujo de control

Temas: 4.2 Categorías de técnicas de diseño de pruebas

- Técnica de caja blanca (“white box”)



Temas: 4.2 Categorías de técnicas de diseño de pruebas

Modelos de caja negra y caja blanca

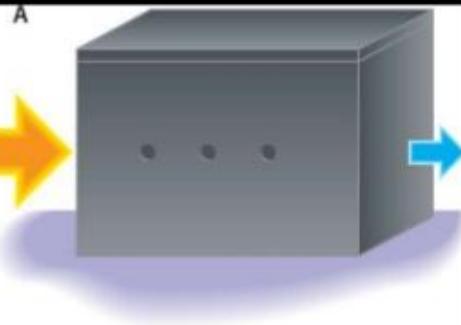
Es otra forma de utilizar modelos, atendiendo a lo que ocurre en el interior del sistema.

Modelo de caja negra

Interesan sólo las entradas y salidas de materia, energía e información en el sistema, y no los elementos e interacciones que suceden en el interior.

Modelo de caja blanca

Se tienen en cuenta las entradas y las salidas, así como las interacciones, las conexiones interiores y las relaciones entre los posibles subsistemas.



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

Métodos de caja negra:

- Participación equivalente o clase de equivalencia
- Análisis de valores límites
- Tablas de decisión y gráficos causa-efectos
- Pruebas de transición de estado
- Pruebas de caso de uso
- Pruebas estáticas
- Pruebas smoke test



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra



4.3.1 Participación de equivalencias o clases de equivalencias (CE)

- Las pruebas se ejecutan utilizando un único representante de cada CE
- Las clases de equivalencia se encogen para entradas (inputs) válidas y no válidas
- Ejemplo:

Si el valor x se define como $0 \leq X \leq 100$, se pueden identificar tres CE

1. $X < 0$ ➔ valores de entrada no válidos
2. $0 \leq X \leq 100$ ➔ valores de entrada válidos
3. $X > 100$ ➔ valores de entrada no válidos



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- La clase de equivalencia para cada variable (elemento) será dividida adicionalmente:
 - ✓ CE válida: todos los valores pertenecientes al rango de definición se combinan en una única clase de equivalencia si son tratados de forma idéntica por el objeto de prueba
 - ✓ CE no válida: se distinguen dos casos que se encuentran fuera del rango de definición.
 - Valores con un formato correcto pero fuera del rango
 - Valores expresados en un formato erróneo pero construido en un CE independiente

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

Ejemplo 1:

Un programa espera el valor de un peso expresado en Kg

- ✓ Todos los valores X^3 son válidos y todos los valores negativos o no numéricos son no válidos.

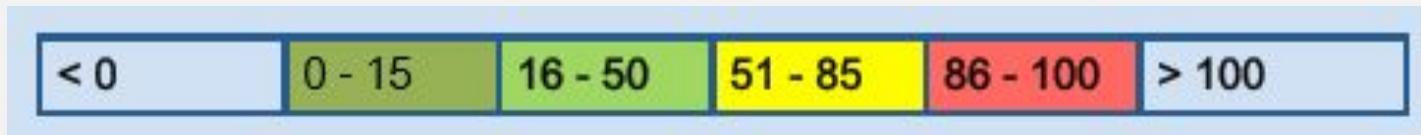
- Clase de equivalencia no valida $X < 0$
- Clase de equivalencia no valida $X = \text{no numérico}$
- Podemos tener otras clases inválidas:
 - ✓ $X > 0$, pero mayor que el nro. más grande que el sistema involucrado puede aceptar
 - ✓ $X > 0$, pero menor que el menor de los nro. que el sistema involucrado puede aceptar

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- Ejemplo 1:
- El porcentaje será presentado en un diagrama de barra. Se aplicarán los siguientes requisitos adicionales (ambos valores incluidos):
 - Valores entre 0 y 15: barra color gris
 - Valores entre 16 y 50: barra color verde
 - Valores entre 51 y 85: barra color amarillo
 - Valores entre 86 y 100: barra color rojo

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- ✓ El Ahora hay cuatro clases de equivalencia válidas en lugar de una:
 - 1ra clase de equivalencia válida: $0 \leq x \leq 15$
 - 2da clase de equivalencia válida: $16 \leq x \leq 50$
 - 3ra clase de equivalencia válida: $51 \leq x \leq 85$
 - 4ta clase de equivalencia válida: $86 \leq x \leq 100$



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- ✓ En el último paso, se determina un representante de cada clase de equivalencia así como el resultado esperado para cada uno de ellos

Variable	Clase de equivalencia	Representantes
Valor porcentaje (válido)	EC ₁ : $0 \leq x \leq 15$	+ 10
	EC ₂ : $16 \leq x \leq 50$	+ 20
	EC ₃ : $51 \leq x \leq 85$	+ 80
	EC ₄ : $86 \leq x \leq 100$	+ 90
Valor porcentaje (no válido)	EC ₅ : $x < 0$	-10
	EC ₆ : $x > 100$	+200
	EC ₇ : x no entero	1,5
	EC ₈ : x non numérico	fred

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- ✓ La cobertura de clases de equivalencia pueden ser utilizadas como criterio de salida para finalizar las actividades del proceso de pruebas.

$$\text{Cobertura (CE)} = \frac{\text{Número de CE probados}}{\text{Número de CE definidos}} * 100 \%$$

- ✓ Beneficios:
 - Método sistemático para el diseño de casos de prueba
 - La partición del rango de valores en CE a partir de las especificaciones cubre los requisitos funcionales

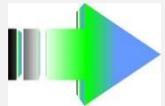
Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

✓ Beneficios:

- La asignación de prioridades a las CE puede ser utilizada para la asignación de los casos de prueba
- Las pruebas de excepciones conocidas están cubiertas por los casos de prueba de acuerdo con las clases de equivalencia negativa



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra



4.3.2 Análisis de valores límite

El análisis de valores límite supone que:

- La clase de equivalencia está compuesta de un rango continuo de valores
- Se pueden definir los límites para el rango de valores
- Como extensión a la partición en CE, el análisis de valores límite es un método que sugiere la selección de representantes
 - ❖ Partición en clases de equivalencias:
 - ✓ Evalúa un valor típico de la CE
 - ❖ Análisis de valores límites
 - ✓ Evalúa los valores límites (frontera) y su entorno

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- ❑ ¿Porqué prestar más atención a los límites?
 - ✓ Frecuentemente los límites del rango de valores no están bien definidos o conducen a distintas interpretaciones
 - ✓ Comprobar si los límites han sido implementados (programados) correctamente
 - ✓ El análisis de los valores límite puede ser aplicado en todos los niveles de prueba. Es fácil de aplicar y su capacidad de detección de defectos es alta en el caso de especificaciones detalladas

¡La experiencia demuestra que, con mucha frecuencia, los errores tienen lugar en los límites del rango de valores!



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

Definición de valores límite

- ✓ El esquema básico sólo puede ser aplicado cuando el rango de valores ha sido definido de conformidad con el mismo esquema.
- ✓ En este caso no son necesarias pruebas adicionales para un valor en el interior del rango de valores

Rango de valores: $\text{Valor Máximo} \leq x \leq \text{Valor Mínimo}$		
Valor Mínimo - δ	Límite Inferior	Límite Inferior + δ
Valor Máximo - δ	Límite Superior	Valor Máximo + δ
δ es el menor incremento definido para el valor. Por ejemplo: 1 para valores enteros.		

- ✓ Si una CE está definido como un único valor numérico, por ejemplo, $x = 5$, los valores correspondientes al entorno también serán utilizados
 - Los representantes (de la clase y su entorno) son: 4,5 y 6



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

Ejemplo:

- ✓ Rango de valores para un descuento en %: $0,00 \leq X \leq 100,00$
- ✓ Definición de CE:
 - 3 clases:
 - 1. CE: $X < 0,00$
 - 2. CE: $0,00 \leq X \leq 100,00$
 - 3. CE: $X > 100,00$
- ✓ Análisis de valores límite
 - Extiende los representantes a:
 - 2. EC: -0,01; 0,00; 0,01; 99,99; 100,00; 100,01
- ✓ En lugar de un representante para la CE válida, ahora hay seis representantes (cuatro válidos y dos no válidos)

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

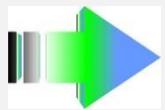
□ Ejemplo:

✓ Esquema básico:

- Seleccionar tres valores con el objeto de ser probados
 - ❖ El valor límite exacto y dos valores pertenecientes al entorno (dentro y fuera de la CE)
- Punto de vista alternativa:
 - ❖ Dado que el valor límite pertenece a la CE, solo son necesarios dos valores para las pruebas, uno perteneciente al CE y otro no perteneciente al CE



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra



4.3.3 Pruebas de tabla de decisión ("decision table testing")

- ✓ La partición en clases de equivalencia y el análisis de valores límite tratan **entradas** en condiciones **aisladas**
- ✓ Sin embargo, una condición de entrada puede tener efectos sólo en **combinación** con otras condiciones de entrada
- ✓ Todos los métodos descritos previamente no tienen en cuenta el efecto de **dependencias y combinaciones**
- ✓ El uso del conjunto completo de las combinaciones de todas las clases de equivalencia de entrada conduce, normalmente, a un número muy alto de casos de prueba (explosión de casos de prueba)
- ✓ Con la ayuda del **gráfico causa-efecto** ("cause-effect graph") y tablas de decisión obtenidas a partir de aquellos, la cantidad de combinaciones posibles se puede reducir de forma sistemática a un subconjunto de las mismas



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- Ejemplo 1: Banca Online (“Online-Banking”)

		T01 T02	T03	T04	T05
Precondiciones (Causas)	Suficiente cobertura	SI	NO	-	-
	Receptor correcto	SI	-	NO	-
	TAN válido	SI	-	-	NO
Actividades (Efectos)	Realizar transferencia	SI	NO	NO	NO
	Marcar TAN como utilizado	SI	NO	NO	NO
	Denegar transferencia	NO	SI	SI	NO
	Solicitar TAN nuevamente	NO	NO	NO	SI

- Cada columna de la tabla representa un caso de prueba

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

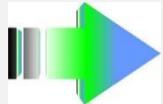
Ejemplo 1: Banca Online (“Online-Banking”)

➤ Construcción de la tabla de decisión:

- Seleccionar un efecto
- Retroceder en el diagrama para identificar la causa
- Cada combinación de causas está representada por una columna en la tabla de decisión (un caso de prueba)
- Combinaciones de causas idénticas, conducentes a efectos distintos, se pueden fusionar para formar un único caso de prueba



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra



4.3.4 Pruebas de transición de estado

- ❖ Muchos métodos sólo tienen en cuenta el comportamiento del sistema en términos de **datos de entrada** (input data) y **datos de salida** (output data)
- ❖ No se tiene en cuenta los diferentes estados que pueda tomar el objeto de prueba
- ❖ Los distintos estados que pueden tomar un objeto de prueba se modelan a través de diagramas de transición de estado
- ❖ El **análisis de la transición de estado** se utiliza para definir casos de prueba basados en a transición de estados

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

Árbol de transición de estado

- ❖ Para determinar los casos de prueba utilizando un diagrama de transición de estado se construye un **árbol de transición**
 - ✓ El estado inicial es la raíz del árbol
 - ✓ Para cada estado que pueda ser alcanzado desde el estado inicial, se crea un nodo que está conectado a la raíz por una rama
 - ✓ Para todos los estados, las posibles transiciones de un estado a otro se representan como ramas



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

Árbol de transición de estado

- ✓ Esta operación se repite y finaliza cuando:
 - El estado del nodo es un estado final (una hoja del árbol)
 - El mismo nodo con el mismo estado ya es parte del árbol
- ✓ **Cada camino** desde la raíz a una hoja entonces representa un caso de prueba de prueba de transición de estado
- ✓ El árbol de transición de estado para este ejemplo conduce a los siguientes seis **casos de prueba**

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

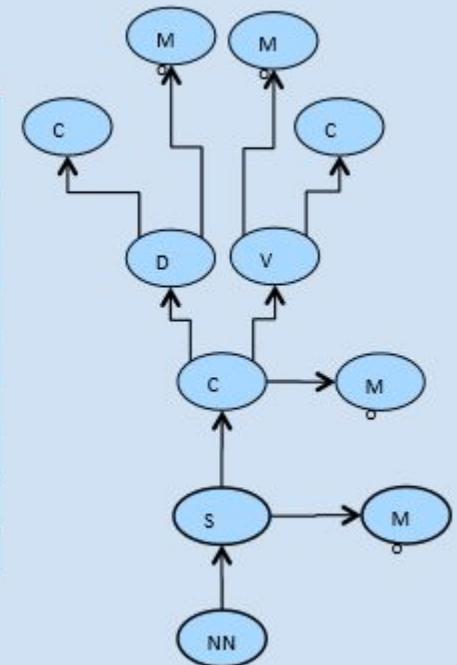
Árbol de transición de estado:



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

El árbol de transición para la pila tiene la siguiente estructura:

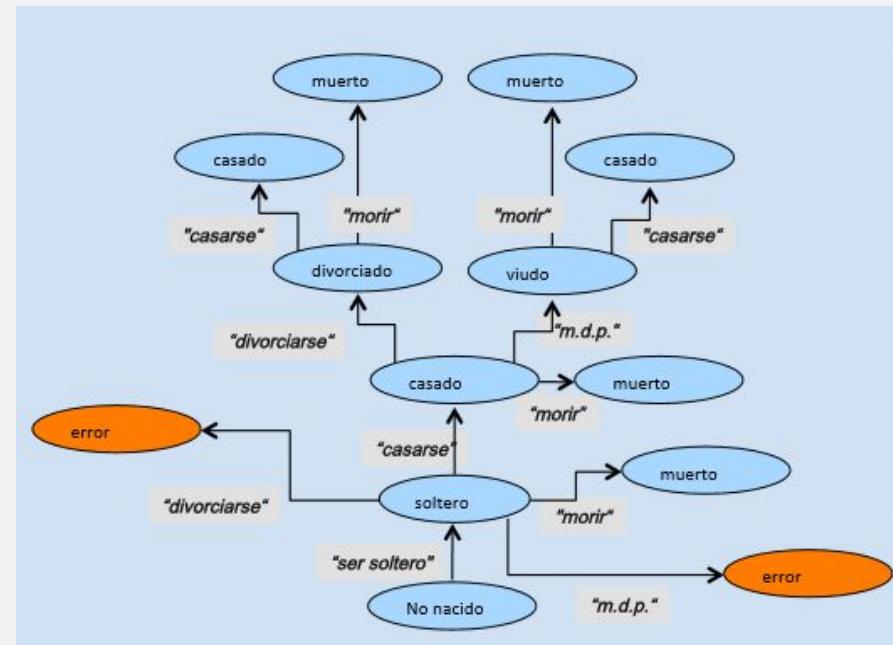
estado 1	estado 2	estado 3	estado 4	estado 5	estado final
no nacido	soltero	muerto			muerto
no nacido	soltero	casado	muerto		muerto
no nacido	soltero	casado	viudo	muerto	muerto.
no nacido	soltero	casado	viudo	casado	casado
no nacido	soltero	casado	divorciado	muerto	muerto
no nacido	soltero	casado	divorciado	casado	casado



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

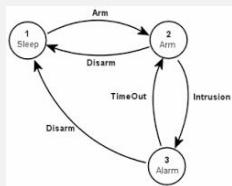
Árbol de transición de estado:

- El árbol de transición de estado de nuestro ejemplo puede ser utilizando transiciones Inválidas (casos de prueba negativos, prueba de robustez)
- Ej: dos transiciones inválidas posibles
- Las transiciones imposibles entre estados no se pueden probar



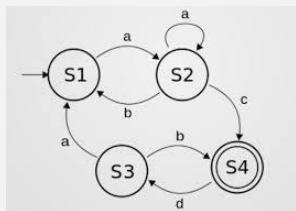
Resumen: Pruebas de transición de estados

- ✓ Criterio de salida de la prueba:
 - **Cada estado** debe haber sido alcanzado al menos una vez
 - **Cada transición** debe haber sido ejecutada al menos una vez
- ✓ Beneficios /desventajas de este método:
 - Buen método de prueba para aquellos objetos de prueba que pueden ser descritos como **una máquina de estado**
 - Buen método de pruebas para probar clases, solo en el caso de disponer del **ciclo de vida del objeto**

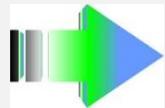


Resumen: Pruebas de transición de estados

- Con frecuencia, los estados son más bien complejos, es decir, es necesario un gran número de parámetros para describir el estado
 - ✓ En estos casos, diseñar casos de prueba y analizar los resultados de las pruebas puede ser difícil y puede implicar un consumo de tiempo considerable
- La sola cobertura de todos los estados no garantiza una cobertura completa de las pruebas



Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra



4.3.5 Pruebas de caso d uso

- Los casos de prueba se obtienen directamente a partir de los casos de uso del objeto de prueba
 - ✓ El objeto de prueba es visto como un **sistema** reaccionando con **actores**
 - ✓ Un caso de uso describe la interacción de todos los actores involucrados conduciendo a un resultado final por parte del sistema
 - ✓ Todo caso de uso tiene **precondiciones** que deben ser cumplidas con el objeto de ejecutar el caso de uso (caso de prueba) de forma satisfactoria
 - ✓ Todo caso de uso tiene **pos condiciones** que describen el sistema tras la ejecución del caso de uso (caso de prueba)

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- Los casos de uso son elementos del Lenguaje Unificado de Modelado (Unified Modeling Language – UML)
- ✓ El **diagrama de casos de uso** es uno de los 13 diferentes tipos de diagramas utilizados por UML
- ✓ Un diagrama de casos de uso describe un comportamiento, no describe una secuencia de eventos
- ✓ Un diagrama de casos de uso muestra la reacción del sistema desde el **punto de vista del usuario**

UML: es un lenguaje de especificación no propietario para el modelo de objetos

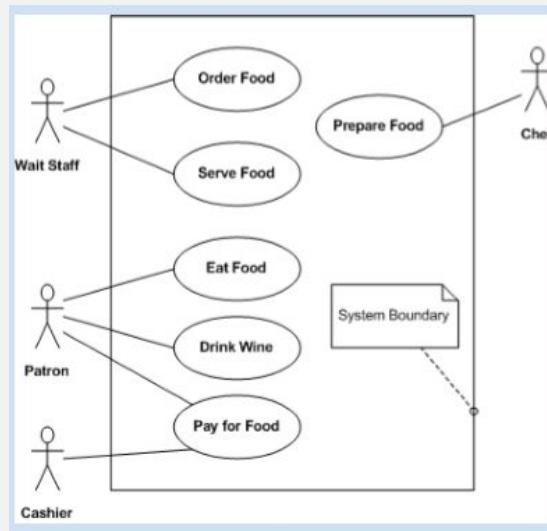
Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

- Cada caso de uso describe una cierta tarea (interacción usuario-sistema)
- La descripción de un **caso de uso** incluye, pero no está limitado a:
 - ✓ Precondiciones
 - ✓ Resultados esperados / comportamientos del sistema
 - ✓ Pos condiciones
- Estos elementos descriptivos también son utilizados para definir el **caso de prueba** correspondiente
- Cada **caso de uso** puede ser utilizado como la fuente para un **caso de prueba**
- **Cada alternativa** en el diagrama corresponde a un **caso de prueba separado**
- Normalmente la información aportada por un caso de uso **no tiene suficiente detalle** para definir casos de prueba. Son necesarios datos adicionales (datos de entrada, resultados esperados) para construir/desarrollar un caso de prueba

Temas: 4.3 Técnicas basadas en la especificación o técnicas de caja negra

➤ Ejemplo:

- ✓ El diagrama describe la funcionalidad de un Sistema “Restaurant”
- ✓ Los casos de uso están representados por óvalos y los actores están representados por las figuras
- ✓ La caja define los límites del sistema, es decir, los casos de uso representados son parte del sistema a modelar y no los actores



Resumen:

✓ Beneficios:

- Pruebas apropiadas para pruebas de aceptación y pruebas de sistema, dado que cada caso de uso describe un escenario de usuario a probar
- Pruebas apropiadas si las especificaciones del sistema se encuentran disponibles en UML



✓ Desventajas:

- Nula obtención de casos de pruebas adicionales mas allá de la información aportada por el caso de uso
- Este método debería ser utilizado solo en combinación con otros métodos de diseño sistemático de casos de pruebas



Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

Las pruebas basadas en la estructura se basan en una estructura identificada del software o sistema:

- ✓ *Nivel de componente*: sentencias, decisiones, ramas o incluso caminos distintos
- ✓ *Nivel de integración*: árbol de llamadas
- ✓ *Nivel de sistema*: estructura de menús, procesos de negocio o una estructura de página web



Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

- Pruebas de sentencia y cobertura (statement testing and coverage)
- Pruebas de rama (branch) y cobertura (branch testing and coverage)
- Pruebas de decisión y cobertura (decision testing and coverage)
- Pruebas de camino (path) y cobertura (path testing coverage)
- SLYSC (LCSAJ- Linear Code Sequence and Jump): Secuencial Lineal y Salto de Código
- Técnicas basadas en el flujo de datos



Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

- ❖ Estas técnicas representan las técnicas de pruebas dinámicas mas importantes y utilizadas de forma mas frecuente.
- ❖ Estas técnicas están relacionadas con las técnicas de análisis estático
- ❖ Durante las pruebas de caja blanca, el programa objeto de las pruebas es ejecutado de la misma forma que las pruebas de caja negra.
- ❖ Ambas categorías (caja blanca y caja negra) conforman las pruebas dinámicas

Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

- ❖ El grado de cobertura de un programa se mide con el uso de herramientas (analizadores de cobertura)
- ❖ La instrumentación del código se lleva a cabo con el objeto de contar la ejecución de caminos
- ❖ Estos contadores son inicializadores a cero, cada ejecución del camino específico incrementara el contador correspondiente

“Los contadores que mantienen el valor cero tras las pruebas indican las partes del programa que aun no han sido ejecutadas”

Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

Principales tipos de cobertura:

- ✓ Cobertura de sentencia (statement coverage)
- ✓ Cobertura de decisión (cobertura de rama) (decision coverage = branch coverage)
- ✓ Cobertura de camino (path coverage)
- ✓ Cobertura de condición (condition coverage)



Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

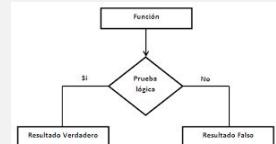
4.4.1 Pruebas de sentencias y cobertura

- ✓ En las pruebas de componente, la cobertura de sentencia es la evaluación del porcentaje de sentencias ejecutables que han sido practicadas por un juego de casos de prueba.
- ✓ Esta determinada por el numero de sentencias ejecutables cubiertas (diseñadas o ejecutadas) por casos de prueba dividido entre el numero de todas las sentencias ejecutables en el código objeto de la prueba

Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

4.4.1 Pruebas de sentencias y cobertura

- ✓ La medición de la cobertura se realiza con el uso de herramientas diseñadas de forma específica.
- Estas herramientas son:
 - Herramientas de Análisis de cobertura (Coverage Analysis Tool)
 - Analizadores de cobertura (Coverage Analyzers)
- Beneficios/Desventajas de este método:
 - Sera detectado el código muerto
 - No podrán ser detectadas: instrucciones faltantes



Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

4.4.2 Pruebas de decisión y cobertura

- ❖ La cobertura de decisión, asociada a las pruebas de rama, es la evaluación del porcentaje de los resultados de decisión que han sido practicados por un juego de casos de prueba
- ❖ Las ramas forman puntos de decisión en el código
- ❖ La cobertura de decisión viene determinada por el número total de resultados de decisión cubiertos (diseñados o ejecutados) por caso de prueba dividido entre el número total de posibles resultados de decisión en el código objeto de la prueba
- ❖ La cobertura de decisión es más fuerte que la cobertura de sentencia, un 100 % de cobertura de decisión garantiza un 100 % de cobertura de sentencia, pero no al contrario

Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

4.4.3 Otras técnicas basadas en la estructura

- Existen niveles más fuertes de cobertura estructural:
 - ✓ Cobertura de decisión
 - ✓ Cobertura de condición
 - ✓ Cobertura de condición múltiple
- La cobertura se puede aplicar a otros niveles de prueba
- El uso de herramientas constituye un soporte útil para las pruebas estructurales del código

Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

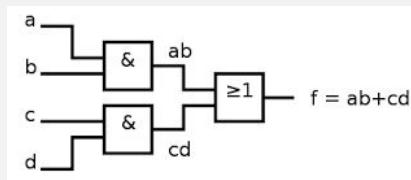
Cobertura de condición:

- Se tienen en cuenta la complejidad de una condición que este constituida por múltiples condiciones atómicas
- Tiene por objetivo detectar defectos que resulten de la implementación de condiciones múltiples (condiciones combinadas)
- Hay tres tipo de cobertura de condición:
 - ✓ Cobertura de condición simple (simple condition coverage)
 - ✓ Cobertura de condición múltiple (multiple condition coverage)
 - ✓ Mínima cobertura de condición múltiple (mínimum multiple condition coverage)

Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

Cobertura de condición:

- Las condiciones múltiples están constituidas por condiciones atómicas, que se combinan con el uso de operadores lógicos como: OR, AND, XOR, etc. Ej: $((a>2) \text{ OR } (b<6))$
- Las condiciones atómicas no contienen operadores lógicos, solo contienen operadores relacionales y el operador NOT ($=, >, <$, etc)



Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

Cobertura de camino (path coverage)

- Se centra en la ejecución de todos los posibles caminos a través de un programa
- Un camino es una combinación de segmentos de programa de flujo de control: secuencia de nodos y aristas alternadas
- Para cobertura de decisión, un solo camino a través de un bucle es suficiente. Para la cobertura de camino hay casos de prueba adicionales
- Esto puede conducir a un numero muy alto de casos de prueba

Temas: 4.4 Técnicas basadas en la estructura o técnicas de caja blanca

Cobertura de camino – Conclusiones:

- El 100 % de cobertura de camino solo se puede lograr en programas muy simples
- La cobertura de camino es más exhaustiva que la cobertura de sentencia y de decisión, cada posible camino a través del programa es ejecutado
- 100 % de cobertura de camino incluye 100 % de cobertura de decisión, que a su vez incluye 100 % de cobertura de sentencia



Resumen:

- ✓ Los métodos de caja blanca y caja negra son métodos dinámicos, el objeto de prueba es ejecutado durante las pruebas
- ✓ El método de caja blanca comprende: cobertura de sentencia, cobertura de decisión, cobertura de camino y cobertura de condición simple, múltiple y mínimo múltiple
- ✓ Solo se puede probar código existente
- ✓ Los métodos de caja blanca son utilizados en pruebas de bajo nivel como pruebas de componentes o pruebas de integración
- ✓ Los métodos difieren en la intensidad de las pruebas (profundidad de las pruebas)

Temas: 4.5 Técnicas basadas en la experiencia

- ❖ Las pruebas basadas en la experiencia son aquellas en las que las pruebas se derivan de la habilidad e intuición del probador y de su experiencia con aplicaciones y tecnologías similares
- ❖ Esta técnica puede tener distintos grados de efectividad, en función de la experiencia del probador
- ❖ Una técnica basada en la experiencia es la predicción de error y pruebas exploratorias
 - ✓ Predicción de error: consiste en enumerar una lista de posibles defectos y diseñar pruebas para atacar dichos defectos
 - ✓ Este enfoque sistemático se denomina ataque de faltas

Temas: 4.5 Técnicas basadas en la experiencia

- ❖ Las pruebas exploratorias coinciden con las fases de diseño de pruebas, ejecución de pruebas, registro de pruebas y aprendizaje
 - ✓ Se trata de un enfoque útil en los casos en los que las especificaciones son escasas o inadecuadas y existe una importante presión temporal o para aumentar o complementar otras pruebas más formales
 - ✓ Puede servir como comprobación del proceso de pruebas, para ayudar a garantizar que los defectos más graves han sido efectivamente detectados

Temas: 4.6 Selección de técnicas de prueba

□ La selección depende de una serie de factores entre los que se incluyen:

- ✓ el tipo de sistema
- ✓ los estándares normativos
- ✓ el nivel de riesgo
- ✓ el objetivo de prueba
- ✓ el tipo de riesgo
- ✓ la documentación disponible
- ✓ los requisitos contractuales o de cliente
- ✓ el conocimiento de los probadores
- ✓ el tiempo y el presupuesto
- ✓ el ciclo de vida de desarrollo
- ✓ los modelos de caso de uso
- ✓ la experiencia previa en los tipos de defectos detectados



Temas: 4.6 Selección de técnicas de prueba

- Algunas técnicas resultan más aplicables a ciertas situaciones y niveles de prueba, mientras que otras son aplicables a todos los niveles de pruebas
- Para crear casos de prueba, los probadores aplican una combinación de técnicas de pruebas, entre las que se incluyen técnicas guiadas por procesos, reglas y datos, con vistas a garantizar la correcta cobertura del objeto que se está probando





¡Muchas Gracias!