Calidad del software

Unidad I
Introducción a la calidad del software

1

Motivación al estudio de la calidad del software

- Los clientes se vuelven mas selectivos y comienzan a rechazar los productos poco fiables o que no dan respuesta a sus necesidades.
- Necesidad de sistemas confiables, que cumplan los requerimientos establecidos.

2

Estudio de la calidad

Calidad del PRODUCTO

Son diferentes

Calidad del PROCESO de desarrollo.

3

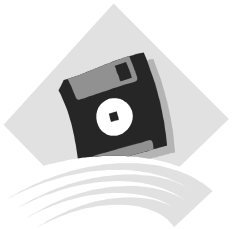
Características especiales del software

- Es un producto MENTAL, sin límites por leyes de la física. Es ABSTRACTO y su calidad también lo es.
- Se DESARROLLA, no se fabrica. El costo está fundamentalmente en el proceso de diseño y no en la producción.

4

Características especiales del software (continuación)

- NO SE DETERIORA con el tiempo. Sus fallos son diferentes a los del hardware. Todos los problemas durante el mantenimiento estaban allí desde el principio y afectan a todas las copias del mismo; no se generan nuevos errores.
- Es ARTESANAL en gran medida.



5

Características especiales del software (continuación)

- El mantenimiento de Software es mas complejo que el mantenimiento de Hardware.
- Es engañosamente fácil realizar cambios sobre un producto de software: Los efectos de los cambios se propagan de forma explosiva e incontrolada.

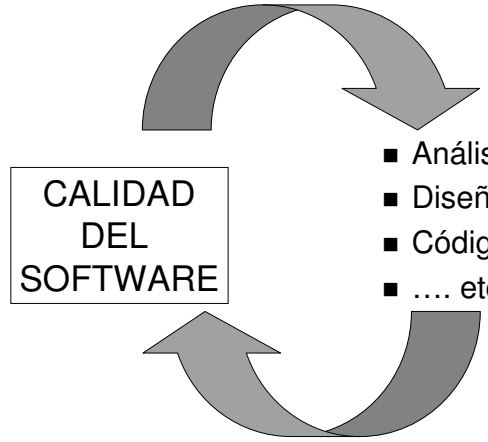
6

Características especiales del software (continuación)

- Desarrollo de software es aun joven. Las técnicas también por lo que no son totalmente efectivas o no están totalmente calibradas.
- El software con errores NO se rechaza. Se asume que es inevitable que el software presente errores.

7

INVOLUCRA TODOS SUS ESTADOS DE EVOLUCION



8

"…El problema de la gestión de la calidad no es lo que la gente NO sabe sobre ella, el problema es lo que creen que saben…

A este respecto, la calidad tiene mucho en común con el sexo: Todo el mundo lo quiere (bajo ciertas condiciones, por supuesto). Todo el mundo cree que lo conoce (incluso aunque no quiera explicarlo). Todo el mundo piensa que su ejecución solo es cuestión de seguir las inclinaciones naturales (Después de todo, nos las arreglamos de alguna forma). Y por supuesto, la mayoría de la gente piensa que los problemas en estas áreas están producidos por otra gente (Como si solo ellos se tomaran el tiempo para hacer las cosas bien)…"

Philip Crosby. Libro: Quality Is free. Mc Graw Hill, 1979.

9

Problemas al abordar tema "Calidad de Software"

- Definición misma de "calidad"
- Comprobación de la calidad
  - *Control de Calidad.- Actividades para evaluar la calidad de los productos desarrollados.*
- Mejora de la calidad
  - *Gestión de Calidad.- Determinación y aplicación de las políticas de calidad de la empresa.*
  - *Garantía o aseguramiento de la calidad.- Actividades planificadas y sistemáticas para proporcionar confianza en que el software satisfará los requisitos dados de calidad.*

10

Definiciones de Calidad

- "Conformidad con los requisitos y confianza en el funcionamiento" [Deming]
- "Adecuación para su uso" [Juran]
- "Hacerlo bien a la primera" [Crosby]
- "Capacidad del producto software para satisfacer los requisitos establecidos" [DoD 2168]

11

Definiciones de calidad

(Continuación)

- "Suma de todos aquellos aspectos o caracteristicas de un producto o servicio que influyen en su capacidad para satisfacer las necesidades expresadas o implícitas" [ISO 8402]
- "Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas" [IEEE 729-83]

12

CALIDAD DEPENDE DE Requisitos que se desea satisfacer

REQUISITOS PREESTABLECIDOS

COMPARACION

PRODUCTO REALMENTE DESARROLLADO

13

REQUISITOS

REQUISITOS EXPLICITOS

✓

QUIERO QUE ESTE PROCESO SEA… Y TENGA… Y YO OBTENGA…

REQUISITOS IMPLICITOS

✗

SE LO QUE QUIERO, PERO NO SE COMO EXPRESARLO

14

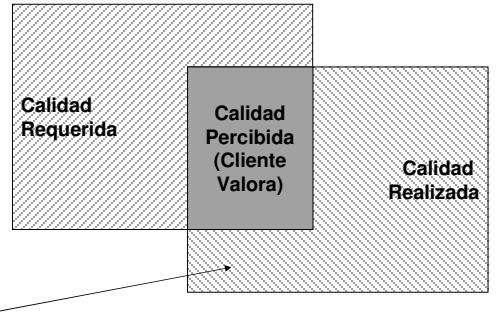
Visiones de la calidad

- **Necesaria o Requerida.-** La que quiere el cliente
- **Programada o Especificada.-** La que se ha especificado explícitamente y se intenta conseguir.
- **Realizada.-** La que se ha conseguido.

OBJETIVO:
Que Coincidan las 3 visiones.

15

Visiones de la Calidad



CUIDADO:
Calidad Realizada pero innecesaria = Gasto inútil de tiempo y dinero

16

Modelos de calidad de software

- Ayudan a definir la calidad del software de forma precisa y útil.
- Ayudan en la puesta en práctica del concepto general de calidad.
- Convierten la calidad en algo concreto, que se puede definir, medir y planificar.
- Ayudan a comprender las relaciones que existen entre diferentes características del software.
- No ha sido demostrada su validez absoluta.

17

Forma jerárquica de la definición de Calidad con los Modelos

Calidad del Software

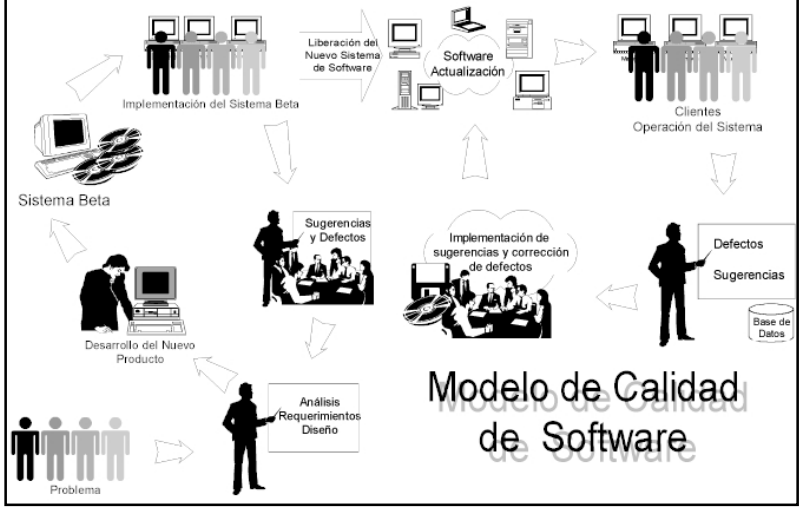
Factores de Calidad — Punto de Vista del Usuario (Atributos de calidad Externos)

Criterios de Calidad del producto — Punto de Vista del Software (Atributos de calidad Internos)

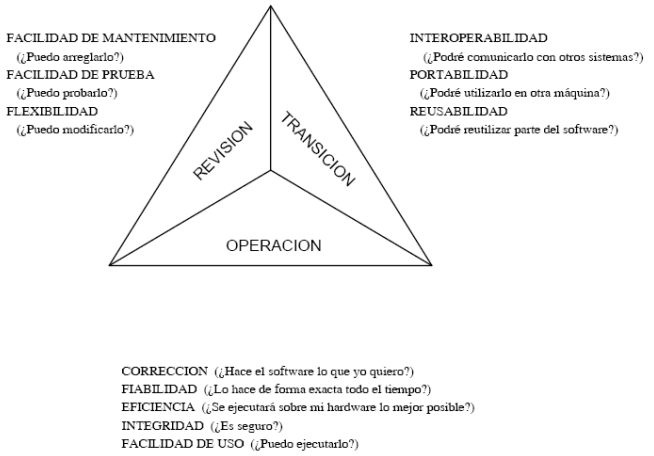
Métricas del producto — Medidas cuantitativas que indican el grado de calidad que posee un atributo.

18

Ejemplo



Modelo de McCall



CORRECCION (¿Hace el software lo que yo quiero?)
FIABILIDAD (¿Lo hace de forma exacta todo el tiempo?)
EFICIENCIA (¿Se ejecutará sobre mi hardware lo mejor posible?)
INTEGRIDAD (¿Es seguro?)
FACILIDAD DE USO (¿Puedo ejecutarlo?)

Modelo de McCall

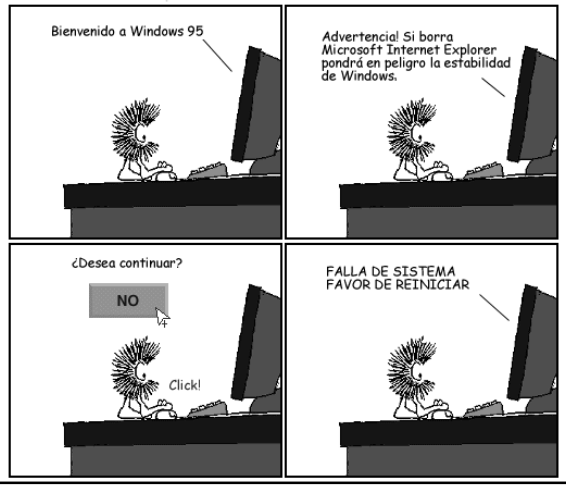
Punto de vista	Factores
Operación del producto	▪Facilidad de uso
	▪Integridad
	▪Corrección
	▪Fiabilidad
	▪Eficiencia
Revisión del producto	▪Facilidad de Mantenimiento
	▪Facilidad de prueba
	▪Flexibilidad
Transición del producto	▪Facilidad de reutilización
	▪Interoperabilidad
	▪Portabilidad

21

Punto de vista	Factores	Criterios
Operación del producto	Facilidad de uso	•Facilidad de operacion •Facilidad de Comunicación •Facilidad de Aprendizaje
	Integridad	•Control de Accesos •Facilidad de auditoría
	Corrección	•Completitud •Consistencia •Trazabilidad
	Fiabilidad	•Precisión •Consistencia •Tolerancia a fallos •Modularidad •Simplicidad
	Eficiencia	•Eficiencia en ejecución •Eficiencia en Almacenamiento

22

Facilidad de uso, Integridad, corrección, fiabilidad, eficiencia



23

Facilidad de uso, Integridad, corrección, fiabilidad, eficiencia

24

Punto de vista	Factores	Criterios
Revisión del producto	Facilidad de Mantenimiento	•Modularidad •Simplicidad •Consistencia •Concisión •Auto descripción
	Facilidad de prueba	•Modularidad •Simplicidad •Auto-descripción •Instrumentación
	Flexibilidad	•Auto-descripción •Capacidad de expansión •Generalidad •Modularidad

25

Punto de vista	Factores	Criterios
Transición del producto	Facilidad de reutilización	•Auto-descripción •Generalidad •Modularidad •Independencia entre sistema y software •Independecia del hardware
	Interoperabilidad	•Modularidad •Compatibilidad de comunicaciones •Compatibilidad de datos
	Portabilidad	•Auto-descripción •Modularidad •Independencia entre sistema y hardware •Independencia del hardware

26

Ejemplo: “Completitud” dentro del factor “Corrección”

Pregunta	Requerimientos	Diseño	Implementación
No hay referencias ambiguas			
Todas las referencias a datos definidas son calculadas u obtenidas de una fuente externa			
Todas las funciones definidas son utilizadas			
Todas las referencias a funciones están definidas			
Se han definido todas las condiciones y procesamientos para cada punto de decisión			
Concuerdan todos los parámetros de llamada a funciones definidos y referenciados			
Todos los informes de problemas se han resuelto			
El diseño concuerda con los requisitos			
El código concuerda con el diseño			
Numero Total de Respuestas “SI”			
	Num de SI / 6	Num de SI / 8	Num de SI / 8
		Suma de las 3 columnas / 3	

Consejo para utilizar el Modelo de McCall

- Para que las métricas se puedan usar sin problemas se recomienda normalizar sus valores en el intervalo [0 , 1]

28

Otras métricas

Fiabilidad = 1 – (Num de errores / Num lineas de codigo)

Facilidad de Mantenimiento = 1 – 0.1 * (Numero medio de diasHombre por Corrección)

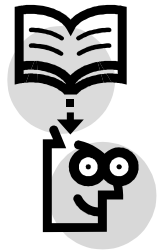
Portabilidad = 1 – (Esfuerzo para portar / esfuerzo para implementar)

Flexibilidad = 1 – 0.05 * (Num medio de diasHombre por cambio)

29

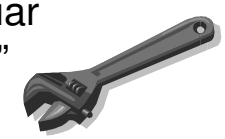
Otros indicadores para evaluar la "Fiabilidad" de un programa

- Num de errores en el programa
- Num de errores en la documentación
- Num de problemas que han aparecido / meses de uso
- Porcentaje de usuarios con problemas



30

Otros indicadores para evaluar "Facilidad de mantenimiento"



- Probabilidad de que un la causa de un incidente sea diagnosticado (o un fallo sea corregido).
- Tiempo medido de reparación o cambio
- Numero de problemas sin resolver
- Tiempo empleado en problemas sin resolver
- Porcentaje de cambios que introducen defectos
- Número de módulos afectados por cada cambio

31

Estrategias de uso de un modelo de calidad

- Modelo fijo
- Definición particular de calidad

32

- MODELO FIJO
  - Se aceptan los factores, criterios y métricas que propone el modelo
  - Se aceptan las relaciones entre factores y criterios y entre criterios y métricas.
  - Solo es necesario seleccionar un subconjunto de los factores de calidad como requisitos de calidad para el proyecto

33

- DEFINICION PARTICULAR DE CALIDAD
  - Se acepta la filosofía de la descomposición.
  - Se selecciona un subconjunto de los factores de calidad como requisitos de calidad para el proyecto.
  - Se decide la descomposición mas adecuada para los factores de calidad seleccionados.

34

Pasos para el uso de un modelo de calidad [al principio del proyecto]

- Seleccionar cuales de los factores de calidad van a ser requisitos de calidad en el sistema.
- Organizarlos en orden de importancia.
- El modelo proporciona los criterios asociados a esos factores.
- Para cada criterio se eligen las métricas.
- Se establecen valores deseables en función de datos históricos.
- Se establecen valores mínimos aceptables.
- La explicación de las decisiones realizadas se debe documentar.

Consideraciones al seleccionar los factores de calidad que serán “requisitos de calidad”

- La relación que tienen los factores con las características peculiares del proyecto.

Si se espera que…	Optar por este Factor…
El ciclo de vida del sistema sea largo	■Facilidad de mantenimiento ■Flexibilidad
Las especificaciones cambien frecuentemente	■Flexibilidad
El hardware evolucione rápidamente	■Portabilidad
Algunas funciones del sistema se usen por largo periodo de tiempo	■Facilidad de reutilización

36

Consideraciones al seleccionar los factores de calidad que serán "requisitos de calidad" [continuación]

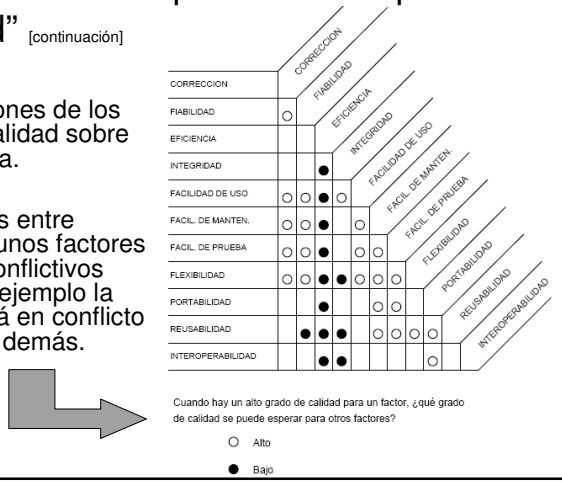
- El coste del factor de calidad frente al beneficio que proporciona

Factor	Ahorro que se puede esperarsi se consigue:
Corrección	■Alto
Fiabilidad	■Alto
Eficiencia	■Bajo
Integridad	■Bajo
Facilidad de uso	■Medio
Facilidad de Mantenimiento	■Alto
Facilidad de prueba	■Alto
Flexibilidad	■Medio
Portabilidad	■Medio
Reusabilidad	■Medio
Interoperabilidad	■Bajo

37

Consideraciones al seleccionar los factores de calidad que serán "requisitos de calidad" [continuación]

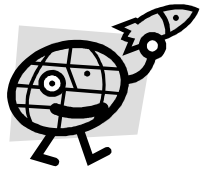
- Las implicaciones de los factores de calidad sobre el ciclo de vida.
- Las relaciones entre factores. Algunos factores pueden ser conflictivos entre sí. Por ejemplo la Eficiencia está en conflicto con todos los demás.



Pasos para el uso de un modelo de calidad [durante el desarrollo]

- Implementar las métricas.
- Analizar los resultados de las métricas.
- Tomar medidas correctivas si es necesario

(Si valores obtenidos < valores mínimos aceptables)



39

Pasos para el uso de un modelo de calidad [al final del proyecto]

- Validar las medidas predictivas utilizadas y comprobar si en efecto se pueden tomar como indicadores de los valores finales

40

Desventajas de los modelos de calidad

- Requieren planificación detallada y una cuidadosa recolección de medidas.
- Puede ser costoso incluso para un numero reducido de factores.
- Requieren recursos adicionales.

- Por estas razones muchas veces se adopta una visión de la calidad mas restringida llamada “Visión simplista”

41

Visión Simplista de la Calidad

- Se basa en el numero de fallos es decir, numero de Defectos conocidos.

Densidad de Defectos = Numero de Defectos / Tamaño del producto

- Oscila según la industria entre 2 y 60 KNCSS

KNCSS = Kilo Non Comment Source Statements
KNCSS = Mil Instrucciones de código fuente sin comentarios

42

Visión simplista de la calidad

- Puede ser usada también en el análisis y diseño, tomando una definición de defecto adecuada, basada en el número de cambios requeridos.

- PELIGRO : Puede ser mal utilizada

43

Consideraciones al utilizar la visión simplista de la calidad

- No todos los defectos conducen a un fallo.
- Solo los fallos son percibidos por el usuario (inciden en la calidad).
- “ 1/3 de los defectos totales conducen a fallos que solo ocurren en promedio cada 5000 años de tiempo de ejecución o mas ”.
- “Solo un 2 % de los defectos son responsables de los fallos que ocurren cada 5 años o menos”

44

Consideraciones al utilizar la visión simplista de la calidad [continuación]

- Un producto que casi no falla puede tener muchos defectos…El usuario los percibe como de alta calidad…
- La visión simplista puede reflejar más el proceso de detección de defectos que la calidad del producto.
- Un producto con pocos defectos detectados puede indicar un proceso de pruebas poco exhaustivo y con defectos ocultos.

45

Terminología

- **ERROR.-** Incorrección cometida por un humano durante el proceso de desarrollo.
- **DEFECTO.-**
  - Consecuencia de un error
  - Desviación en el valor esperado.
  - No afecta el funcionamiento del sistema.
- **FALLO.-** Manifestación de un defecto en el software.
- **FALLAS.-** Defectos aún no detectados.
- **INCIDENTE.-** Situación en la que se produce y se informa de un comportamiento no esperado del sistema.

46

Ejercicio

- ¿Cree Ud. Que se podría encontrar una única fórmula matemática que permitiese calcular el grado de calidad de cualquier producto de software?
- Elija 3 de entre las 11 características de calidad del modelo de McCall y sugiera 2 posibles métricas que se podrían utilizar para evaluar dicha característica.

47

Calidad del software

Unidad II
Actividades de Control
de Calidad del Software

48

Objetivo de las actividades de control de calidad

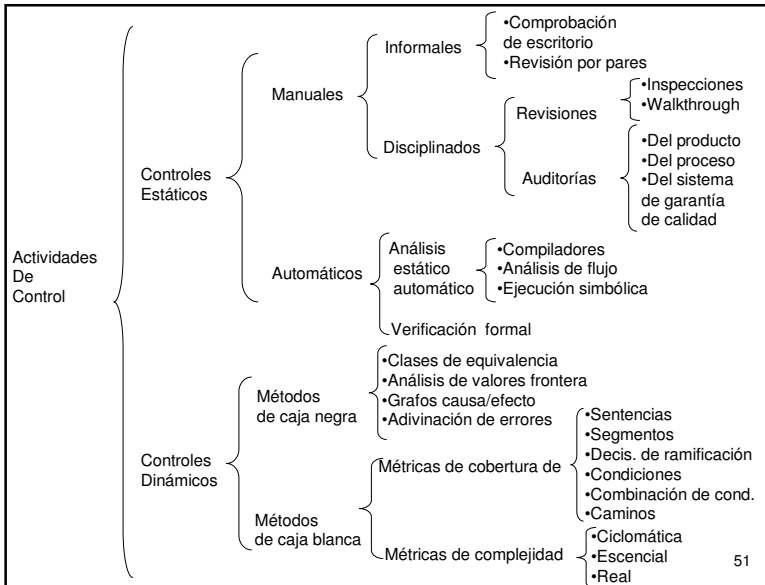
- Comprobar si un producto posee o no una determinada característica de calidad en el grado requerido.
- Cuando un producto no posee una determinada característica de calidad se dice que tiene un DEFECTO.
- Puede decirse que el Objetivo del Control de Calidad es identificar defectos y corregirlos.

49

Actividades de Control de Calidad

- Controles Estáticos.- Analizan el objeto sin necesidad de ejecutarlo.
- Controles Dinámicos.- Requieren la ejecución del objeto que está siendo probado.

50



51

Controles Estáticos – Manuales - Informales

- **Comprobación de Escritorio.-** (Desk checking) Consiste en examinar a mano e individualmente el objeto que se acaba de desarrollar. Es el método mas tradicional para analizar un programa.
- **Revisión por pares.-** (Peer view) Consiste en la revisión del código de un programador por otros programadores.

52

Controles Estáticos – Manuales - Disciplinados

- Revisiones
  - Inspecciones
  - Walkthrough
- Auditorías
  - Del producto
  - Del proceso
  - Del sistema de garantía de calidad

53

Revisiones

- Reunión formal en la que se presenta el estado actual de los resultados de un proyecto a un usuario, cliente u otro tipo de persona interesada, y se realiza un análisis estructurado de los mismos.
- La evaluación técnica no recae sobre las mismas personas involucradas en la producción del software.
- Tipos de Revisiones:
  - Inspecciones
  - Walkthrough (Visita guiada)

54

"...El trabajo técnico necesita ser revisado por la misma razón que los lápices necesitan gomas de borrar: Errar es humano. La segunda razón por la que necesitamos revisiones técnicas es que, aunque la gente es buena descubriendo algunos de sus propios errores, algunas clases de errores se le pasan por alto mas fácilmente al que los origina que a otras personas. El proceso de revisión es, por lo tanto, la respuesta a la plegaria de Robert Burns:

'Qué gran regalo sería poder vernos como nos ven los demás!'

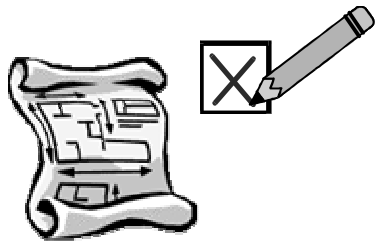
..."

Freedman y Weinberg

55

Inspecciones

- Los participantes van leyendo el documento paso a paso, guiados por el autor del mismo y comprobando en cada paso el cumplimiento de los criterios en una lista de comprobación.

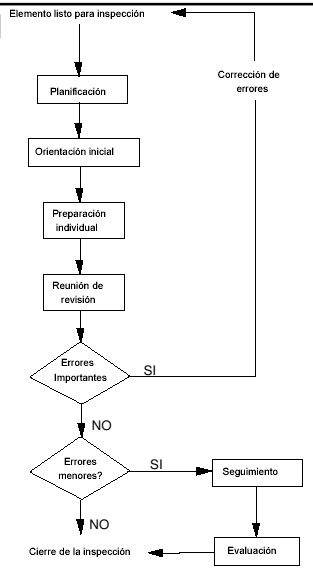


56

Fases en una inspección

Al planificar una inspección también se debe determinar:

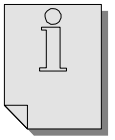
- Los objetivos de la inspección
- Los criterios de finalización
- El lugar y la fecha
- Disponibilidad de todos los participantes
- La agenda de la reunión



57

Documentos generados en una inspección

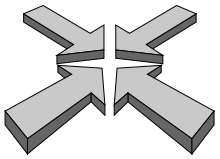
- **Informe resumen de la inspección.-**
  - Conclusiones breves para la dirección
  - Que se revisó, quien lo hizo y la conclusión.
- **Lista de acciones pendientes.-**
  - Para los autores del producto. No llega a la dirección
  - Explica lo que está mal y como corregirlo.
  - Debe ser Claro y sencillo.
- **Informe de asuntos relacionados.-**
  - Problemas relacionados INDIRECTAMENTE con el objeto revisado.
- **Informe del proceso de inspección.-**
  - Su algo sale mal en el proceso de inspección en sí mismo.
- **Informe final.-**
  - Se informa a la dirección del cierre de la inspección.



58

Consejos para una inspección exitosa

- Revisar el producto, no al productor.
- Fijar una agenda y mantenerla.
- Limitar el debate y las impugnaciones.
- Enunciar áreas de problemas, pero no intentar resolver cualquier problema que se ponga de manifiesto.
- Tomar notas escritas.
- Limitar el número de participantes e insistir en la preparación anticipada.
- Desarrollar una lista de comprobación.
- Disponer recursos y una agenda.
- Llevar a cabo un bien entrenamiento de los revisores.
- Repasar las revisiones anteriores,



59

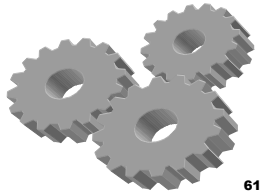
Walkthrough (Visita Guiada)

- Se demuestra la funcionalidad del objeto revisado mediante la simulación de su funcionamiento con casos de prueba y ejemplos.
- Se introducen los casos de prueba y se van registrando los resultados intermedios

60

Fases en un walkthrough

- **Planificación.-** Similar a la planificación de una inspección, con la diferencia de que no es necesario asignar roles específicos a los participantes, a excepción del presentador, que es quien organiza y guía la reunión.
- **Preparación individual.-** Cada revisor examina el objeto revisado. En este caso no hay lista de comprobación.
- **Reunión del walkthrough**



61

Diferencias entre Inspecciones y Walkthrough

- ¿A quién ayudan?
  - Walkthrough.- Al desarrollador
  - Inspecciones.- Al Gestor
- ¿Cuál es el objetivo?
  - Walkthrough.- Entendimiento y comprensión del objeto.
  - Inspecciones.- Detectar defectos
- ¿Quién guía el proceso?
  - Walkthrough.- La estructura del objeto revisado
  - Inspecciones.- Una lista de comprobación (CheckList)
- ¿Qué tanta preparación requieren?
  - Walkthrough.- No tan planificadas.
  - Inspecciones.- Se planifican mas formalmente

62

Diferencias entre Inspecciones y Walkthrough

Propiedades	Inspección	Walkthrough
1. Requiere entrenamiento formal del moderador	SI	NO
2. Hay unos roles definidos para los participantes	SI	NO
3. Quién guía la revisión	El moderador	El propietario del producto revisado
4. Se usan listas de comprobación	SI	NO
5. Se usan distribuciones por tipo de los errores a buscar	SI	NO
6. Hay un seguimiento para controlar que la corrección es correcta	SI	NO
7. Se puede mejorar la eficiencia de la revisión analizando los resultados	SI	NO

63

Formalidad en las revisiones (Inspecciones y walkthrough)

- Formales
  - Evento público.
  - Se informa por escrito de los resultados.
  - Todos son responsables de la calidad de la revisión.
  - Ventajas:
    - Resultados son Exitos para el proyecto.
    - Al ser público, se promueve una mejor preparación de los participantes.
  - Desventajas:
    - Son Impersonales
- Informales

64

Revisiones técnicas y Revisiones de gestión (de proyecto)

- Tipos de Revisiones técnicas
  - Revisión de la especificación de los requisitos.
  - Revisión del diseño.
  - Revisión del código.
  - Revisión de las pruebas.
  - Revisión del manual de usuario.

- Objetivos de las Revisiones de gestión
  - Control de la progresión del proyecto
  - Evaluación de los riesgos asociados al proyecto
  - Evaluación general del producto

65

Controles Estáticos - Automáticos

- Análisis Estático Automático
  - Compiladores
  - Análisis de flujo
  - Ejecución simbólica
- Verificación Formal

66

Análisis Estático Automático : - Compiladores

- Pueden detectar:
  - Expresiones sintácticamente incorrectas
  - Incompatibilidades de tipo
  - Errores de Tipo semántico
  - Etc.

67

Análisis Estático Automático : - Análisis de flujo

- Representación gráfica (Grafos)
  - Nodos: sentencias o segmentos de programa
  - Arcos: Posibles transiciones de control desde un segmento a otro
- Identifica caminos, analiza comportamiento, sitúa puntos de ruptura.
- Técnica de Coke y Allen sigue la secuencia del valor de las variables en cada instrucción ejecutada. Las clasifican en:
  - Referenciadas [R] .- (Parte derecha de una asignación, o se accesa)
  - Definidas [D] .- (Parte izquierda de una asignación)
  - No referenciadas [N] .- (Variables locales fuera de la subrutina)
  - No Afectadas [I] .- No modifica su valor

68

Análisis Estático Automático : - Análisis de flujo [continuación]

- OJO con :
  - **…D D…** (Se ha definido dos veces sin ser referenciada).
  - **…N R…** (Se ha referenciado sin haberla definido previamente).
  - **R…** (Se ha referenciado sin haberla definido previamente).
- Ejemplo:

```
A = X + 1
A = 3
MsgBox str(A)    ……  etc….
```

Para la variable "A" : D D R …
Para la variable "X" : R I I …

69

Análisis Estático Automático : - Ejecución Simbólica

- Consiste en la ejecución simbólica de ciertos caminos dentro del programa, durante la cual se verifican ciertas expresiones simbólicas con respecto a ciertas condiciones y afirmaciones preestablecidas.
- Solo se usan símbolos y nombres de variables.
- El resultado de la ejecución simbólica se compara con la especificación externa del programa y se verifica que coincidan.
- Se puede usar un árbol para representar puntos de decisión.
- Desventajas:
  - Programa sin ciclos = Arbol finito. Programa con ciclos = Arbol infinito
  - Crea indecisibilidad

70

Análisis Estático Automático : - Ejecución Simbólica [continuación]

- Ejemplo:

Función en Ada que calcula la suma de los N elementos de un array entero A:

```
Function SUM (A: INTARRAY; N: NATURAL) return INTEGER is
  S: INTEGER := 0;
  I: NATURAL := 1;
Begin
  while I <= N loop
    S := S + A(I);
    I := I + 1;
  end loop;
  return (S);
End SUM;
```

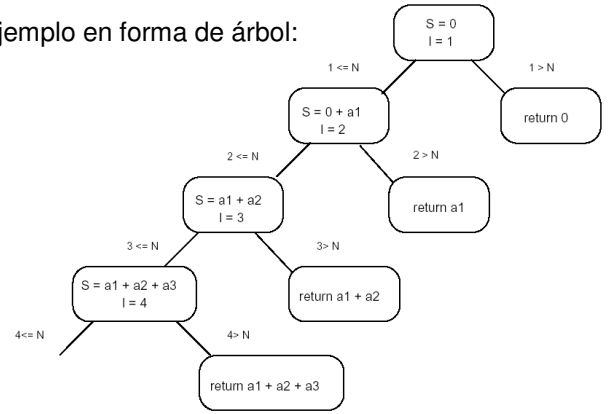
Resultado de la ejecución simbólica:

$$S = \sum_{j=1}^{I-1} a_j$$ sustituyendo I por N+1 $$S = \sum_{j=1}^{N} a_j$$

71

Análisis Estático Automático : - Ejecución Simbólica [continuación]

- Ejemplo en forma de árbol:



72

Análisis Estático Automático : - Verificación formal

- Demostración matemática de la corrección de un programa con respecto a sus especificaciones.
- Se considera el programa como una cadena de un lenguaje formal, con sintaxis y semántica formal.
- NO siempre es posible realizar este tipo de verificación.
- SOLO se utiliza para sistemas críticos debido al costo alto que conlleva.

73

Controles dinámicos

- Requieren la ejecución del objeto que se está probando o de un modelo del mismo.
- **Prueba del software.-** Proceso en el que se ejecuta un sistema con el objetivo de detectar fallos.
- **Depuración.-** Proceso de localización, corrección y evaluación del efecto de la correción de un Defecto que causa un fallo.
- El costo de detección de defectos es mas alto que el Costo de corrección de los mismos.
- El proceso de prueba puede toma hasta el 50% o 60% del esfuerzo dedicado al proyecto.

74

Tipos de pruebas

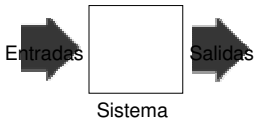
De acuerdo al IEEE 1012-1986 el conjunto mínimo de pruebas que se deben realizar son:

- **Prueba modular, unitaria o de Componentes.-** Se prueba cada módulo aislado del resto.
- **Prueba de integración.-** Se prueba la inter-relación entre los módulos.
- **Prueba del sistema.-** Comprueba que el sistema satisface los requisitos del usuario.
- **Prueba de aceptación.-** Se realiza ya que el sistema se implantó. Demuestra al usuario que el sistema satisface sus necesidades.
- **Prueba de regresión*.-** Comprueba que toda nueva versión de un software es de no menos calidad que la versión anterior.

75

Controles Dinámicos - Métodos de prueba

- **Métodos de Caja Negra.-** (Pruebas funcionales o pruebas orientadas al diseño). Verifican que el sistema cumpla su función sin importar como lo haga.

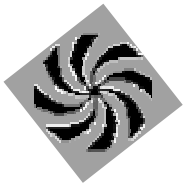


- **Métodos de caja blanca.-** (Pruebas estructurales). Verifican la forma como el sistema cumple con su función por medio de su diseño detallado, diagramas de flujos de datos, control, código fuente, etc.

76

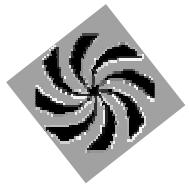
Reflexiones sobre las pruebas

- Un buen caso de prueba es aquel que tiene una probabilidad muy alta de descubrir un nuevo error.
- Una prueba tiene éxito si descubre un nuevo error.
- Una prueba NO asegura la ausencia de defectos. SOLAMENTE puede demostar que existen errores en el software.
- "El 80% de los errores está en el 20% de los módulos". Hay que identificar esos módulos y probarlos muy bien.



77

Reflexiones sobre las pruebas [continuación]

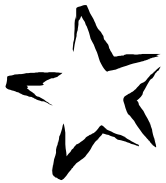


- Las pruebas deben planificarse mucho antes de que empiecen.
- En una prueba es mejor empezar por lo pequeño y progresar hacia lo grande.
- Son mas efectivas las pruebas dirigidas por un equipo independiente.
- TIP: Si se definen casos de prueba con la mayor probabilidad de encontrar el mayor numero de errores, se invierte MENOS cantidad de esfuerzo y tiempo.
- Se recomienda diseñar casos prueba usando Pruebas de caja blanca y caja negra.

78

Una buena prueba…

- Tiene una alta probabilidad de encontrar un error.
- No debe ser redundante.
- Debe ser "La mejor de la cosecha"
- No debe ser ni demasiado sencilla ni demasiado compleja.



79

Métodos de caja negra

- **Método de clases de equivalencia.-** Dividir las entradas en grupos, y cada grupo pruebe las mismas propiedades.
- **Análisis de valores límite.-** Seleccionar datos de entrada que caen en la frontera (justo a un lado, justo al otro y justo en la frontera).
- **Grafos causa/efecto.-** Causas = entradas. Efectos = Salidas. Se construye un grafo para ver cuales causas producen ciertos efectos. Se observan cuales causas producen el mismo efecto.
- **Adivinación de errores.-** Imaginar cuales son los errores que se pudieron cometer con mas probabilidad y generar casos de prueba para comprobar dichos errores.

80

Métodos de caja blanca

- **Basados en métricas de cobertura.-** Se representa el programa mediante un grafo. Nodo = Sentencia(s). Arcos = Flujo. La prueba consiste en probar cada camino posible.
  - Cobertura de sentencias
  - Cobertura de segmentos entre decisiones
  - Cobertura de decisiones de ramificación
  - Cobertura de condiciones
  - Cobertura de todas las combinaciones de condiciones
  - Cobertura de caminos
- **Basados en métricas de complejidad.-** Se usa un grafo para determinar el número de caminos independientes y determinar cuantas pruebas son necesarias para ejecutar cada camino al menos una vez.

81

Cómo hacer las pruebas

1. Planear la prueba
   - Objetivo
   - Objetos a probar
   - Características a probar y cuales no
   - Método de prueba a utilizar
   - Los recursos a emplear
   - Plan de tiempos
   - Productos a generar durante las pruebas
   - El reparto de las responsabilidades.

82

Cómo hacer las pruebas [continuación]

2. Diseñar la prueba.- Instrucciones detalladas acerca de:
   - Como llevar a cabo la prueba
   - De qué forma se van a utilizar los métodos de prueba
   - Qué objetos se van a probar en cada una de las pruebas
   - Qué criterios se van a utilizar para determinar si el objeto pasa o no pasa la prueba.

83

Cómo hacer las pruebas [continuación]

3. Determinar los casos de prueba.- Especificar el conjunto de casos prueba a utilizar. Para cada caso se debe especificar:
   - Qué objetos se van a probar
   - Que entradas se van a dar
   - Cuales son las salidas esperadas

84

Cómo hacer las pruebas [continuación]

4. Planificar el procedimiento de prueba.- Fijar un conjunto de pasos para la ejecución de la prueba. Se especifica detalladamente:
   - Secuencia exacta de ejecución de los distintos casos de prueba.
   - Los requisitos que hay que cumplir para la ejecución de cada caso.
   - Las condiciones de terminación de cada uno de ellos.
5. Ejecución de la prueba / Registro de resultados.
6. Análisis y evaluación de la prueba.

85