

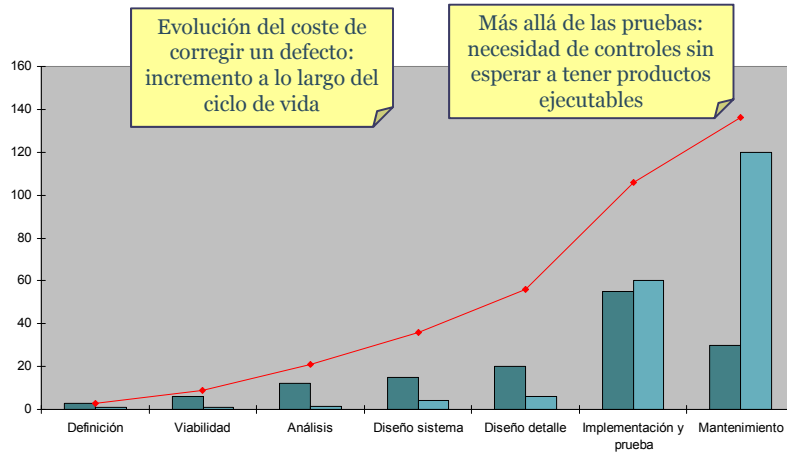
Revisiones del software

Ingeniería del Software Avanzada
Revisiones

V y V

- **Concepto:**
 - Conjunto de procedimientos, técnicas y herramientas
 - Uso paralelo al desarrollo de software
 - Asegurar que el producto satisface necesidades
- **Boehm:**
 - **Verificar:** ¿construir correctamente el producto?
Comprobar coherencia en el ciclo de vida
 - **Validar** ¿construir el producto adecuado? Comprobar si satisface los requisitos

Evolución de costes de corrección



Clasificación de técnicas

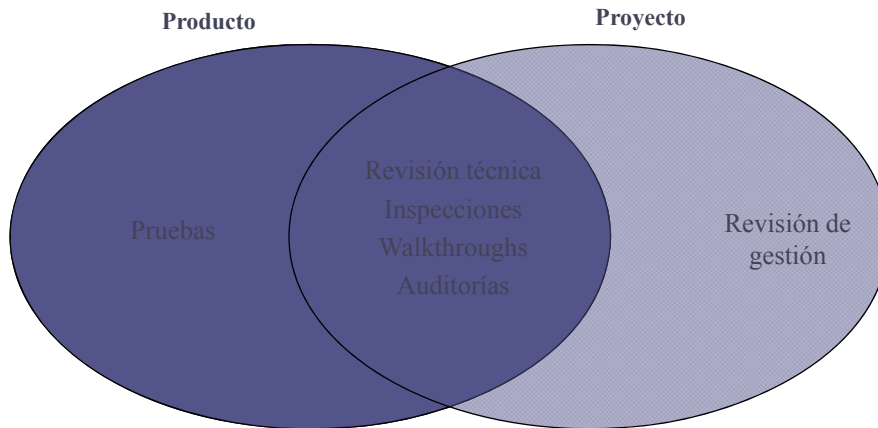
- Revisiones de proyecto
 - Revisiones de gestión
- Revisiones de producto
 - Revisiones técnicas
 - Inspecciones
 - Walkthroughs
 - Pruebas
 - Otras
- Auditorías de software

Dar recomendaciones para actividad de gestión según estado de producto
Control de proyecto
Cambio de dirección de proyecto

Evaluación más o menos formal del software: identifica desviaciones
Satisfacción de especificaciones
Conformidad con planes, estándares, guías aplicables a cada producto

Revisión independiente
Evaluación objetiva de prod. y proc.
Conformidad o implantación: estándares, procedimientos, etc. contractuales o normativos

Grupos



Otras verificaciones y validaciones

- Cada proyecto y organización decidirá las que necesita
- Ejemplos de otras técnicas complementarias:
 - **Análisis de algoritmos.** Verificar funcionalidad y consumo de recursos en tiempo de ejecución.
 - **Análisis de simulación.** Evaluación del rendimiento para planificar la capacidad de un sistema.
 - **Audidores de código.** Examinar código fuente y controlar cumplimiento de estándares y prácticas de programación.
 - **Generadores de referencias cruzadas.** Control basado en nombres de variables, procedimientos, etiquetas, etc.
 - **Analizadores de flujo de control.** Determinar secuencias incorrectas en la ejecución del flujo de control de un programa.
 - **Comprobación de interfaces.** Analizar consistencia y completión y analizar usabilidad y accesibilidad.
 - **Análisis de requisitos.** Errores sintácticos, inconsistencias lógicas o ambigüedades en entradas, salidas, procesos y datos.

Diferentes técnicas

- Tres aspectos que diferencian a las diferentes técnicas de revisión entre sí:
 - Sobre qué se realiza la revisión (producto o proyecto).
 - Objetivos perseguidos (aunque genéricamente sean la búsqueda de defectos).
 - El proceso o mecánica de realización.
- También varía su mecánica:
 - Revisiones informales: no hay procedimientos definidos, se realiza de la forma más flexible posible.
 - Revisiones semi-formales: se definen unos procedimientos mínimos a seguir.
 - Revisiones formales: se define completamente el proceso, los participantes y sus funciones, los documentos, etc.

Comparativa

- Pruebas:
 - Buscan defectos (funcionales y técnicos). No proponen soluciones. Formales o informales, son obligatorias en todo proceso software.
- Revisiones técnicas:
 - Comprobar que el producto diseñado se ajusta a las necesidades del cliente (centradas en análisis y diseño, no programación). No proponen soluciones.
- Revisiones de gestión:
 - Sobre plan de proyecto y su estado, registran desviaciones y proponen acciones, no soluciones a las desviaciones.
- Walkthroughs:
 - Buscan defectos de todo tipo, incluso de estilo. Proponen soluciones.
- Inspecciones:
 - Muy formales y controladas. Buscan defectos de todo tipo, prescindiendo del estilo. No proponen soluciones.
- Auditoría:
 - No buscan defectos de programación. Revisiones independientes, no participativas. Muy formales. Duración en el tiempo (incluso meses). Proponen acciones, no soluciones.

Pruebas - Definiciones

- Pruebas
 - Ejecución de software con datos controlados (casos) con el fin de descubrir defectos
- Caso de prueba
 - Conjunto específico de entrada, procedimientos y salida esperada para una situación de operación
 - Buen caso de prueba, gran probabilidad de detectar defectos no encontrados antes
- Éxito
 - Descubrir un defecto no detectado
- Dijkstra: Las pruebas no pueden asegurar la ausencia de defectos, sólo demostrar que (los que localizamos) existen en el software

Pruebas - Recomendaciones generales

- ☺ Cada caso debe definir en detalle la salida esperada
- ☺ Evitar que el autor pruebe su software (¿no tiene ventajas?)
- ☺ Inspeccionar con detalle la salida obtenida
- ☺ Incluir tanto entradas no válidas e inesperadas como válidas y esperadas
- ☺ Comprobar si el software:
 - No hace lo que debe hacer (fallo de funciones)
 - Hace lo que supone no debe hacer (efectos secundarios)
- ☺ Evitar utilizar casos desechables (control y economía)
- ☺ No planear suponiendo ausencia de defectos
- ☺ Estudios:
 - Probabilidad de nuevos defectos es proporcional al número de defectos ya encontrados

Diseño de pruebas

- Tres enfoques:
 - Funcional o caja negra
 - Particiones de equivalencia
 - Valores límite
 - Conjetura de errores
 - Estructural o caja transparente
 - Grafo de flujo
 - Complejidad ciclomática de McCabe
 - Caminos de prueba ($a-n+2$)
 - Criterios de cobertura (de sentencias, de decisiones,...)
 - Aleatoria

Pruebas - Importancia práctica

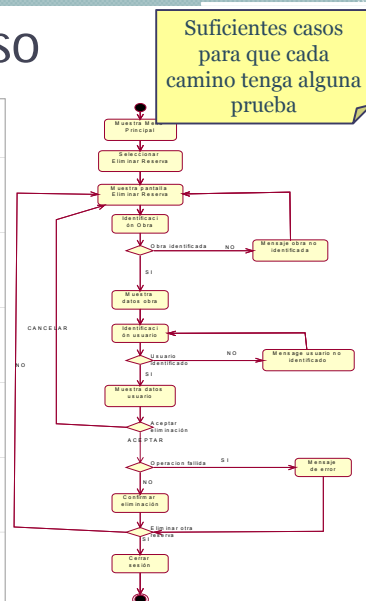
- El diseño de casos es totalmente dependiente de una buena especificación
 - Muchas veces, el trabajo de pruebas supone hacer el trabajo que no hicieron los analistas
- En cuanto tenemos una buena especificación se pueden diseñar los casos de prueba
 - En especial, si contamos con casos de uso
 - No se diseña justo antes de probar
- Eficiencia y limitación de recursos
 - No buscar pruebas perfectas: equilibrio riesgo-coste
- Como última fase, las pruebas sufren los retrasos de todas las fases de desarrollo
 - La planificación de pruebas debe contemplar plazos generosos

Pruebas a partir de casos de uso

- Describen interacciones actor-sistema
 - Eficaces para vincular especificación y pruebas
 - Recogen combinaciones y opciones no válidas (flujos alternativos)
- Generar casos de prueba:
 - Caminos/escenarios de ejecución de caso
 - Diagrama de interacción o de estados
 - Sobre cada escenario, diseño de caja negra:
 - Clases de equivalencia y límites
 - Tratamiento de combinaciones
 - Centrado en datos de E/S manejados en el caso

Ejemplo de caso de uso

Bibliotecario	Sistema
	1.- Muestra por pantalla el menú principal
2.- Selecciona la opción <i>Eliminar Reserva</i>	3.- Muestra la pantalla de <i>Eliminar Reserva</i>
4.- Introduce el <i>idObra</i> <extends Buscar Título >	5.- Muestra los datos de la obra y pide el DNI del usuario.
6.- Introduce el DNI <extends Buscar usuario >	7.- Muestra datos del usuario y confirmar eliminar reserva
8.- Pulsa <i>ACEPTAR</i>	9.- Mensaje de confirmación y si eliminar otra reserva.
10.- Pulsa <i>NO</i>	11.- Cierra sesión y muestra menú principal



Particiones de equivalencia

Añadir valores límite y casos especiales de conjetura de errores

Condición o restricción de entrada	Clases válidas	Clases no válidas
Código de área	(1) $200 \leq \text{código} \leq 999$	(2) código < 200 (3) código > 999 (4) no es número
Texto de la operación	(5) 6 caracteres	(6) < 6 caracteres (7) > 6 caracteres
Operación	(8) ingreso (9) transferencia (10) pago recibo (11) retirada efectivo	(12) ninguna orden válida

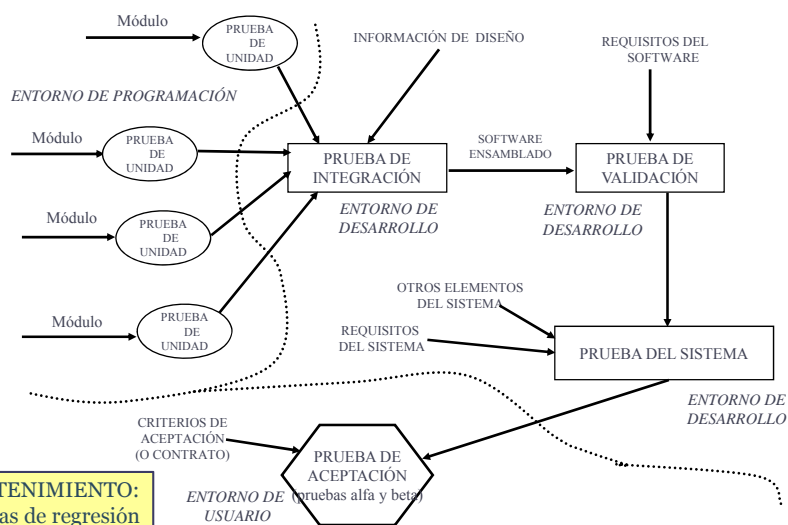
Casos válidos:

- 300 Nómina Ingreso (1)(5)(8)
- 400 Bilbao Transferencia (1)(5)(9)
- 500 I.T.V. Pago recibo (1)(5)(10)
- 600 Cheque Retirada efectivo (1)(5)(11)

Casos no válidos:

- 180 Nómina Ingreso (2)(5)(8)
- 1032 XXXXXX Ingreso (3) (5) (8)
- ZV YYYYYYY Ingreso (4) (5) (8)
- 350 J Transferencia (1)(6) (8)
- 450 Regalos Transferencia (1)(7) (8)
- 550 IRPF97 Saldo (1)(5)(12)

Pruebas - Niveles



Inspecciones

- Técnica de revisión desarrollada por M. Fagan en IBM (1972-5)
- Definición:
 - Técnica de evaluación formal
 - Se examinan requisitos, diseño o código
 - Grupo de personas
 - Para detectar defectos y desviaciones de normas
- Objetivo:
 - Identificar, verificar y registrar defectos
 - No encontrar soluciones o alternativas

Inspecciones - Reglas

- Objetivos: enfocada en descubrir defectos
 - No sobre cómo corregir, añadidos o “mejoras”
- Equipos de inspección
 - Pequeños grupos de compañeros de trabajo (de 3 a 6): 4-5, lo más común.
 - Autor y compañeros que desarrollan productos relacionados
 - Uso de listas de comprobación
 - Duración de reuniones: máximo 2 horas
- Roles
 - Autor no puede ser moderador o lector (presenta el producto)
 - Los moderadores capacitados son esenciales: asegurar preparación, ritmo de reunión, enfoque en defectos y evitar ataques a autor
- Productos
 - Aplicable a muchos productos: especificaciones ,diseño, código, pruebas,..
 - Tamaño habitual: 10-20 páginas o 200-250 LOC
- Salidas
 - Lista de defectos e informe de revisión: qué, quién, nº y gravedad de defectos
 - Tasa de inspección (velocidad en páginas o LOC por hora)

Inspección - Resumen (I)

Etapas	Responsable	Actividades
Planificación	Moderador	Producto cumple requisitos de entrada Conseguir participantes adecuados Fijar lugar y momento adecuados
Vista general (opcional)	Equipo al completo	Formar a participantes Asigna papeles
Preparación	Individual	Cada participante estudia el producto, prepara su papel y anota defectos
Reunión	Equipo al completo	Encontrar defectos (sin discutir posibles soluciones)

Inspecciones - Resumen (II)

Etapas	Responsable	Actividades
Tercera hora	Equipo al completo	Exponer ideas suprimidas en la reunión sobre mejoras o soluciones
Corrección	Autor o a quien se asigne	Corregir defectos encontrados
Seguimiento	Moderador	Verificar la corrección de todos los defectos
Análisis causal	Equipo al completo o depto. de calidad	Analizar estadísticas de reuniones (detectar causas y mejoras en los procesos, métodos desarrollo, etc.)

Inspecciones - Criterios de terminación

- La inspección acaba cuando:
 - Todos los defectos detectados se han resuelto
 - Los resultados de la inspección se han pasado a los informes
- El moderador:
 - Verifica ambos criterios antes de declarar terminada y completa la inspección
- Cada proyecto debería desarrollar criterios propios según las necesidades del mismo y del entorno

Inspecciones - Listas de comprobación

- Series de preguntas o comprobaciones para examinar el producto
- Proporciona definición clara de la tarea a los participantes
- Se construyen a base de acumular experiencias en inspecciones
 - Asimilar las estadísticas de defectos
- Extensión:
 - Una sola página con 20-25 preguntas o ítems
- Pueden usarse en otro tipos e revisiones de software

Walkthroughs

- *Walkthrough*: recorrido del producto
- Revisión de cualquier producto por un grupo de nivel similar al que lo desarrolló
 - En general, del equipo de desarrollo
- Objetivo: el autor explica el producto (presenta el enfoque de diseño y programación e informa a compañeros del progreso de trabajo) mientras los demás se centran en:
 - Encontrar defectos:
 - Errores, omisiones, contradicciones, debilidades
 - Estudio de la técnica y el estilo de desarrollo
 - Búsqueda de alternativas

Walkthroughs - Tipos

- Según el grado de organización y estructuración:
 - Formales:
 - Preparación larga, autor gasta tiempo en presentar, realimentación lenta (semanas, buen trabajo), rápida si es malo.
 - Informales
 - Preparación muy baja, realimentación baja
 - Semiformales
 - Lo más recomendable: ventajas de ambos
- En ciclo de vida:
 - Informales en primeras fases y formales en las finales

Walkthroughs - Roles

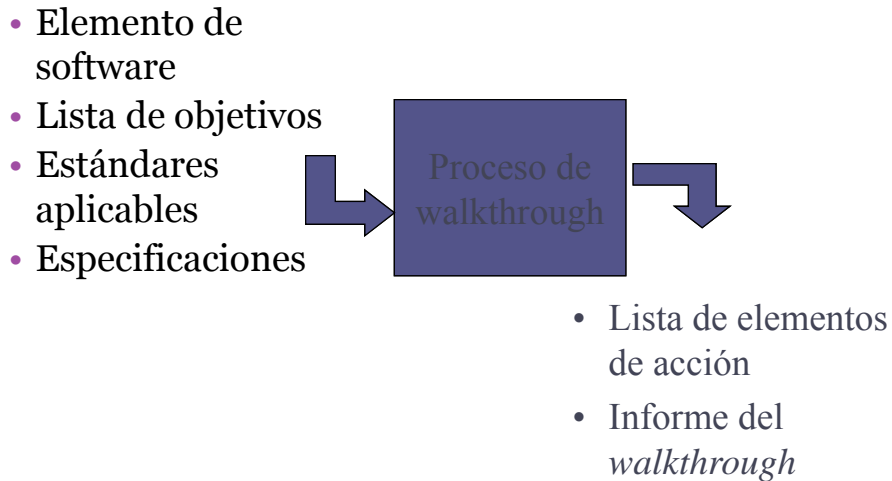
- Autor
- Moderador (puede ser el propio autor)
- Posibles miembros del equipo:
 - Encargado de mantenimiento:
 - Prever futuro de mantenimiento
 - Producto autoexplicativo, mantenimiento sin el autor, etc.
 - Supervisor de estándares
 - Conformidad respecto de normativa y estándares
 - No literal sino el espíritu y teniendo en cuenta el entorno
 - Representante de usuario
 - Ajuste a las necesidades, especialmente en especificaciones
 - Otros revisores, normalmente del mismo nivel que el autor
 - Opinión general sobre corrección y calidad
 - Posibles externos para visión más distanciada

Walkthrough - Tareas

- Reconocimiento de objetivos:
 - Presentador: para guiar la reunión
- Presentación del producto (si procede)
 - Breve, si ha habido suficiente preparación
- Críticas y comentarios
 - Incluyendo walkthroughs anteriores
- Calificación final:
 - Aceptado, aceptado con modificaciones, nuevo walkthrough.

Walkthrough - Documentación

Depende de su grado de formalidad



Revisiones de gestión

- Evaluación formal:
 - Plan de proyecto
 - Estado del proyecto
- Objetivo:
 - Examinar estado de proyecto
 - Acorde con lo esperado del plan
 - Limitado por factores externos
 - Registrar desviaciones
 - Identificar temas importantes a tratar
 - Para estudio de la dirección
 - Para otros responsables u organizaciones implicadas
 - Recomendar acciones a partir de ahora, no soluciones

Revisiones técnicas

- Evaluación formal de un elemento de software por un equipo cualificado de revisión
 - Se ajusta a sus especificaciones
 - Sigue estándares y planes aplicables
 - Implantación adecuada de los cambios
- Tipos:
 - Revisión de requisitos
 - Revisión de diseño
 - Revisión de hitos: ayudan a los gestores a juzgar la compleción de una fase antes de continuar

Auditorías

- Objetivo:
 - Confirmar el cumplimiento de producto y procesos para asegurar el cumplimiento de:
 - Estándares, Guías, Especificaciones, Procedimientos, Ejecución de proyecto y productividad
- Revisión independiente y muy disciplinada
 - No participativa: visión de un lado del asunto
 - Figura de autoridad: auditor
- Permite evaluar:
 - Elementos de software
 - Procesos
 - Proyectos
 - Planes de calidad

Auditorías - ¿Cuándo realizarlas?

- Punto singular del proyecto
 - Seguimiento de planes
 - Fecha, criterio existente, etc.
- Demanda de grupo externo:
 - Agencia reguladora, usuarios o clientes, etc.
 - Requisitos contractuales o normativos
- A petición de la organización:
 - Jefe de proyecto
 - Departamento de SQA

Auditorías - Etapas

- Planificación
 - Objetivos de auditoría y Alcance
- Visión general
 - Definición del proceso de auditoría, calendario y acuerdos de colaboración con la empresa auditada
- Preparación
 - Conocer la organización a auditar y preparar un plan
- Examen
 - Recogida de datos (documentación y entrevistas)
- Análisis
 - Determinar la salud del proyecto o producto usando métricas
- Informe de resultados
 - Defectos, descubrimientos y recomendaciones