

# **9. Garantía de calidad del software**

# Índice

- Referencias
- Introducción
- Conceptos de calidad
  - Calidad.
  - Control de calidad.
  - Garantía de calidad.
  - Coste de calidad
  - El equipo de garantía de calidad del software

# Índice

- Verificación y Validación
  - Introducción
  - El IEEE Std. 1012-2004.
- Revisiones del software
- Revisiones Técnicas Formales
  - Introducción.
  - La reunión de revisión.
  - Registro e informe de revisión.
  - Directrices para la revisión.
  - El IEEE Std. 1028-1997.

# Índice

- Garantía de calidad estadística
  - Mejora estadística.
  - Índice de errores.
- Fiabilidad del software
- Plan SQA. El IEEE Std. 730-2002

# Índice

- El enfoque de calidad ISO 9000
  - Introducción.
  - La familia ISO 9000.
  - Implementación de ISO 9001:2000.
  - Estructura de ISO 9001:2000.
  - AENOR.
- Conclusiones

# Referencias

- Pressman, R.S. *Ingeniería del Software. Un Enfoque Práctico. Sexta Edición.* McGraw-Hill, 2005
- IEEE Std. 730-2002, IEEE Standard for Software Quality Assurance Plans
- IEEE Std. 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

# Referencias

- IEEE Std. 1012-2004. IEEE Standard for Software Verification and Validation
- IEEE Std. 1028-1997, IEEE standard for software reviews
- Software Formal Inspections Guidebook NASA-GB-A302. Software Assurance Technology Center de la NASA, 1993

# Introducción

- Uno de los objetivos principales de la IS es construir software de *calidad*
- Podemos definir *calidad* como una característica o atributo de algo
- ¿Cómo medimos la calidad del software?
- Con medidas indirectas



# Introducción

- Hay dos tipos de calidad:
  - *Calidad de diseño*. Características especificadas para un elemento (software: SRS y Diseño).
  - *Calidad de concordancia*. Grado de cumplimiento de las especificaciones de diseño durante su realización (software: implementación).
- En ingeniería, la garantía de calidad se centra en el *control de variación entre muestras*
- En IS, la variación entre muestras no tiene sentido

# Introducción

- En IS hay dos formas de conseguir calidad:
  - Haciendo SRS, diseños e implementaciones *correctos* desde un punto de vista *técnico*.
  - Introduciendo en el modelo de proceso una serie de actividades que garanticen que *todas* las entregas resultantes de una actividad de trabajo sean *correctas*.
- Las técnicas de IS para conseguir calidad en el software se denominan *Garantía de Calidad del Software* (SQA: *Software Quality Assurance*)

# Introducción

- La SQA engloba:
  - Enfoque de gestión de calidad.
  - Tecnologías de IS (métodos y herramientas).
  - Revisiones Técnicas Formales.
  - Estrategia de pruebas.
  - Control de la documentación y de cambios.
  - Procedimientos que aseguren ajustes a los estándares de IS
  - Mecanismos de medición y generación de informes

# Conceptos sobre calidad

## Calidad

- Definimos *calidad* como *concordancia* con:
  - Los requisitos funcionales y de rendimiento explícitamente establecidos.
  - Los estándares de desarrollo explícitamente documentados.
  - Las características implícitas de todo software desarrollado profesionalmente.

# Conceptos sobre calidad

## Calidad

- El IEEE Std. 610.12 define *calidad* como:
  - Grado en que un sistema, componente o proceso cumple las especificaciones.
  - Grado en que un sistema, componente o proceso cumple las necesidades o deseos de clientes y usuarios.
- Discusión ¿difieren ambas definiciones?

# Conceptos sobre calidad

## Control de calidad

- El *control de calidad* es una serie de inspecciones, revisiones y pruebas utilizados a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados
- Incluye un bucle de realimentación

# Conceptos sobre calidad

## Garantía de calidad

- La *garantía de calidad* es el establecimiento de un marco de procedimientos organizativos que llevan a conseguir una alta calidad del software
- e.g.:
  - El control de calidad nos lleva a hacer una RTF del diseño de una parte del sistema.
  - Hacemos RTFs porque forman parte de la garantía de calidad de nuestra organización.

# Conceptos sobre calidad

## Coste de calidad

- El *coste de calidad* incluye todos los costes que se derivan de la búsqueda de la calidad o en las actividades relacionadas en la obtención de la calidad
- Hay tres tipos de costes de calidad:
  - De prevención.
  - De evaluación.
  - De fallos.



# Conceptos sobre calidad

## Coste de calidad

- Los costes de *prevención* incluyen:
  - Planificación de calidad.
  - Revisiones técnicas formales.
  - Equipo de pruebas.
  - Formación.

# Conceptos sobre calidad

## Coste de calidad

- Los costes de *evaluación* incluyen:
  - Inspección en el proceso y entre procesos.
  - *Calibrado y mantenimiento del equipo.*
  - Pruebas.
- Los costes de *fallos* se dividen en:
  - Internos.
  - Externos.

# Conceptos sobre calidad

## Coste de calidad

- Los fallos *internos* se producen cuando se detecta un error antes de la entrega al cliente.
  - Los costes asociados incluyen:
    - Revisión.
    - Reparación.
    - Análisis de fallos.

# Conceptos sobre calidad

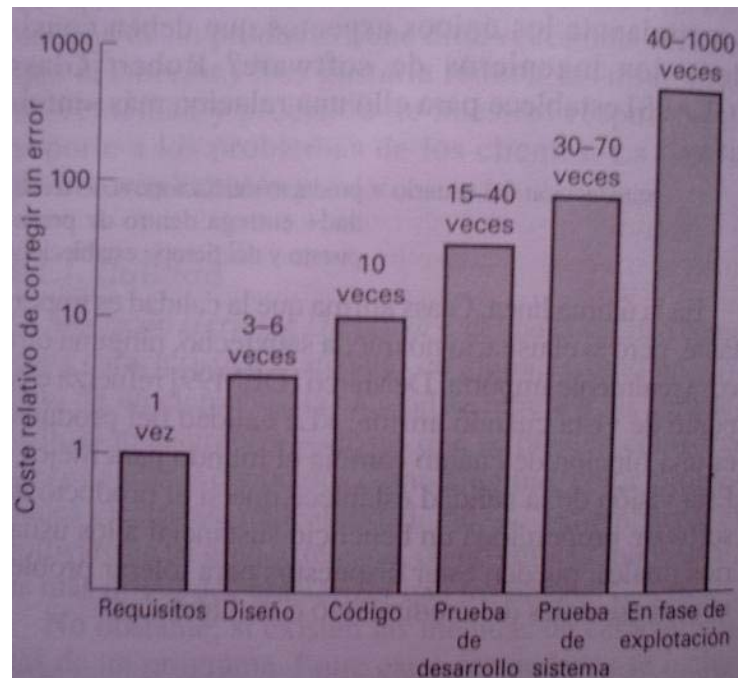
## Coste de calidad

- Los fallos *externos* se producen cuando se detecta un error después de la entrega al cliente
  - Los costes asociados a fallos externos incluyen:
    - Costes de los internos.
    - Resolución de quejas.
    - Devolución y sustitución de productos.
    - Soporte de línea de ayuda.
    - Trabajo de garantía.

# Conceptos sobre calidad

## Coste de calidad

- Evidentemente, cuanto más tardemos en resolver un fallo, más costosa será su resolución



# Conceptos sobre calidad

## Coste de calidad

- Por ejemplo un análisis del trabajo de IBM en Rochester indica:
  - 7053 horas de RTFs de 200000LDC
  - 3112 errores descubiertos
  - Coste programador 40\$/h
  - Coste eliminación defectos: 282120\$ (91\$/defecto)

# Conceptos sobre calidad

## Coste de calidad

- Supongamos que en vez de RTFs somos muy *cuidadosos*
- Supongamos que hay 1 fallo por 1000LDC
- Tenemos unos 200 defectos
- Con un coste de 25000\$ por reparación
- Coste reparación de 5 millones de dólares
- Es decir, 18 veces más caro que el coste de prevención

# Conceptos sobre calidad

## El equipo SQA

- La SQA comprende tareas llevadas a cabo por dos tipos de personal:
  - *Desarrolladores*. Llevan a cabo el trabajo técnico.
  - *Grupo SQA*. Ayudan al equipo de desarrolladores para que el software alcance una alta calidad.



# Conceptos sobre calidad

## El equipo SQA

- Actividades del grupo SQA:
  - Establecimiento del plan SQA para el proyecto.
  - Participar en el desarrollo de la descripción del proceso de software.
  - Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso del software.
  - Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software.

# Conceptos sobre calidad

## El equipo SQA

- Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
- Registrar lo que no se ajuste a los requisitos e informar a sus superiores.
- Coordinar el control y la gestión de cambios.
- Analizar las métricas del software.

# V&V

## Introducción

- Asociado al tema de pruebas y garantía de calidad encontramos *Verificación y Validación (V&V)*
- V&V son los procesos que determinan si los productos desarrollados de una actividad dada se ajustan a los requisitos de esa actividad, y si el software satisface su uso deseado y las necesidades del usuario

# V&V

## Introducción

- Es decir, son procesos que garantizan la calidad del software
- En términos de Boehm:
  - La *verificación* se encarga de comprobar si estamos construyendo el producto correctamente.
  - La *validación* se encarga de comprobar si estamos construyendo el producto correcto.

# V&V

## Introducción

- Según el IEEE Std. 610.12-1990
  - *Verificación* es el proceso de evaluar un sistema o componente para determinar si los productos de una determinada fase de desarrollo satisfacen las condiciones impuestas al comienzo de esa fase
  - *Validación* es el proceso de evaluar un sistema o componente durante o al final del proceso de desarrollo para determinar si satisface los requisitos especificados

# V&V

## Introducción

- *Verificación y validación* es el proceso de determinar si los requisitos para un sistema o componente son completos y correctos, los productos de cada fase de desarrollo cumplen los requisitos o condiciones impuestas por la fase previa, y el sistema o componente final cumple con los requisitos especificados

# V&V

## Introducción

- Dentro del proceso de V&V existen dos aproximaciones complementarias para el análisis y comprobación de los sistemas:
  - Inspecciones/revisiones del software (estáticas).
  - Pruebas del software (dinámicas).

# V&V

## IEEE Std. 1012-2004

- El IEEE Std. 1012-2004. IEEE Standard for Software Verification and Validation define los procesos de V&V en términos de actividades específicas y tareas relacionadas
- También define los contenido del *plan de V&V del software*, incluyendo un ejemplo de formato



# **V&V**

## **IEEE Std. 1012-2004**

- El IEEE Std. 1012-2004 identifica seis procesos a los cuales se puede aplicar V&V:
  - Gestión.
  - Adquisición.
  - Suministro.
  - Desarrollo.
  - Operación.
  - Mantenimiento.

# **V&V**

## **IEEE Std. 1012-2004**

- **Proceso de gestión**
  - Se centra en la gestión de V&V
  - La V&V del proceso de gestión monitoriza y evalúa todas las salidas de V&V

# **V&V**

## **IEEE Std. 1012-2004**

- **Proceso de adquisición**
  - Comienza con la definición de la necesidad de adquirir un sistema, continúa con la preparación y emisión de una petición de propuesta, selección de suministrador y gestión de los procesos de adquisición
  - V&V del proceso de adquisición se encarga del comienzo del proyecto, petición de propuestas, preparación del contrato, monitorización del suministrador y aceptación del sistema

# **V&V**

## **IEEE Std. 1012-2004**

- **Proceso de suministro**
  - Se inicia al generar una respuesta a la petición de propuesta o al negociar, y finaliza con un contrato.  
Continúa con la determinación del plan del proyecto
  - V&V de suministro utiliza los producto del proceso de suministro para confirmar que los requisitos de la petición de propuesta y los requisitos del contrato son consistentes y satisfacen las necesidades del usuario antes de finalizar el contrato

# **V&V**

## **IEEE Std. 1012-2004**

- **Proceso de desarrollo**
  - Contiene las actividades y tareas del desarrollador: análisis de requisitos, diseño, codificación, integración, prueba, instalación y soporte
  - V&V de desarrollo verifica y valida estos productos software

# **V&V**

## **IEEE Std. 1012-2004**

- **Proceso de operación**
  - Implica el uso del sistema software por el usuario final en un entorno operacional
  - V&V de operación evalúa el impacto de cambios en el entorno operativo, valora el efecto de los cambios propuestos en el sistema, evalúa los procedimientos operativos para la observancia del uso deseado, y analiza los riesgos que afectan al usuario y al sistema

# **V&V**

## **IEEE Std. 1012-2004**

- **Proceso de mantenimiento**
  - Se activa cuando el sistema software o la documentación asociada debe cambiarse en respuesta a una necesidad de mantenimiento
  - V&V de mantenimiento valora los cambios propuestos en el sistema y su impacto en el software, evalúa anomalías que son descubiertas durante la operación, valida requisitos de migración, valida la retirada de requisitos y vuelve a llevar a cabo tareas de V&V

# V&V

## IEEE Std. 1012-2004

- A su vez, el estándar identifica cuatro *niveles de integridad*: características del proyecto que determinan el grado de rigor a la hora de aplicar las actividades de V&V
- Los niveles de integridad se asocian a requisitos, funciones, grupos de funciones, componentes o subsistemas



# V&V

## IEEE Std. 1012-2004

- En función del nivel
  - 4: El elemento software debe ejecutarse correctamente u ocurrirán consecuencias graves (pérdida de vidas, pérdidas del sistema o pérdidas sociales). No hay mitigación posible (e.g. control avión)
  - 3: El elemento software debe ejecutarse correctamente o la misión del sistema no se realizará causando consecuencias serias (heridas permanentes, degradación importante del sistema, impacto económico o social). Es posible una mitigación parcial o completa (e.g. banco)

# **V&V**

## **IEEE Std. 1012-2004**

- 2: El elemento software se debe ejecutar correctamente o una función deseada no se realizará, causando consecuencias menores. Una mitigación completa es posible (e.g. videoclub)
- 1: El elemento software se debe ejecutar correctamente o la función deseada no se realizará, causando consecuencias insignificantes. No se requiere mitigación (e.g. procesador textos)

# **V&V**

## **IEEE Std. 1012-2004**

- El estándar también incluye niveles de integridad basados en el riesgo
- En función del nivel de integridad se deberán llevar a cabo distintas tareas de V&V

# Revisiones del software

- Las revisiones del software son un *filtro* para el proceso de IS
- Se aplican en varias etapas del desarrollo del software
- *Purifican* las actividades estructurales
- Hay diversos tipos de revisiones de mayor o menor formalidad y/o valía

# Revisiones del software

- Nosotros nos centraremos en las *Revisiones Técnicas Formales* RTFs o *Inspecciones Formales*
- El objetivo básico de las RTFs es detectar errores antes de que se conviertan en defectos
- Es razonable suponer que un software que no ha pasado por RTFs es susceptible de tener más errores y defectos

# Revisiones Técnicas Formales

## Introducción

- Una RTF es una actividad de garantía de calidad del software llevada a cabo por los desarrolladores
- Los objetivos de la RTF son:
  - Descubrir errores.
  - Verificar que el software alcanza sus requisitos.
  - Garantizar que el software se desarrolla de acuerdo a ciertos estándares predefinidos.

# Revisiones Técnicas Formales

## Introducción

- Conseguir un software desarrollado de manera uniforme.
- Hacer que los proyectos sean más manejables.
- La RTF se instrumenta mediante una reunión, que debe ser convenientemente planificada

# Revisiones Técnicas Formales

## La reunión de revisión

- Normas de la reunión:
  - Deben convocarse para la reunión entre tres y cinco personas.
  - Se debe preparar por adelantado, pero sin que requiera más de dos horas de trabajo por persona.
  - La duración debe ser menor de dos horas.
- Evidentemente, la RTF se centran en partes específicas del software



# Revisiones Técnicas Formales

## La reunión de revisión

- Se deben revisar los productos resultantes de las tareas de trabajo
- El procedimiento es:
  - El responsable del producto (*productor*) informa al jefe del proyecto del fin de un trabajo.
  - El jefe del proyecto contacta con el *jefe de revisión* que distribuye el producto a dos o tres revisores.

# Revisiones Técnicas Formales

## La reunión de revisión

- Cada revisor, incluido el jefe de revisión, revisa el producto durante una o dos horas.
- El jefe de revisor establece una agenda y planifica una reunión (normalmente para el día siguiente).
- A la reunión asisten el productor, los revisores y el jefe de revisión.
- Uno de los revisores juega el papel de *registrador*.

# Revisiones Técnicas Formales

## La reunión de revisión

- El productor expone su producto.
- Los revisores exponen sus pegas.
- Cuando se descubre un problema o error *real*, el registrador toma nota.
- Al final de la reunión se puede decidir:
  - Aceptar el producto sin modificaciones.
  - Rechazar el producto debido a los errores encontrados.
  - Aceptar el producto supuesto que se llevan a cabo algunas modificaciones (sin nueva RTF)

# **Revisiones Técnicas Formales**

## **La reunión de revisión**

- Una vez llevada a cabo la reunión, los participantes firman el registro de revisión
- Así se indica:
  - Asistentes.
  - Conformidad.

# Revisiones Técnicas Formales

## Registro e informe de revisión

- El registrador toma nota de todos los problemas identificados en la RTF
- Al final de la reunión se genera una *lista de problemas de revisión* que resume todas las pegas
- Además genera un *informe sumario de revisión*

# Revisiones Técnicas Formales

## Registro e informe de revisión

- El *informe sumario de revisión* contiene:
  - Qué se revisó.
  - Quién lo revisó.
  - Qué se descubrió, y cuáles son las conclusiones.
- Dicho informe es una página simple (con posibles suplementos) que se adjunta al registro histórico del proyecto

# Revisiones Técnicas Formales

## Registro e informe de revisión

- La *lista de problemas de revisión* sirve para:
  - Identificar las áreas problemáticas dentro de un producto.
  - Servir como *lista de comprobación de puntos de acción* que guíe al productor para hacer las correcciones.
- La lista de problemas puede adjuntarse al informe sumario

# **Revisiones Técnicas Formales**

## **Registro e informe de revisión**

- Cuando el producto es aceptado con cambios, alguien debe encargarse del seguimiento de los cambios identificados en la lista de problemas de revisión
- El revisor jefe o el equipo SQA pueden ser los encargados de dicho seguimiento



# Revisiones Técnicas Formales

## Directrices para la revisión

- Las directrices de revisión deben establecerse de antemano para evitar problemas
- Directrices:
  - 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.

# **Revisiones Técnicas Formales**

## **Directrices para la revisión**

- Tomar notas escritas.
- Limitar el número de participantes e insistir en la preparación anticipada.
- Desarrollar una lista de comprobación para cada producto que vaya a ser revisado.
- Disponer de recursos y una agenda para la RTF.
- Llevar a cabo un buen entrenamiento de todos los revisores.
- Repasar las reuniones anteriores.

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Las nociones sobre RTFs presentadas están inspiradas en el *Software Formal Inspections Guidebook* NASA-GB-A302 del *Software Assurance Technology Center\** de la NASA
- Este manual data de 1993 e incluye listas de comprobación

\*<http://satc.gsfc.nasa.gov/>

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- El IEEE Std. 1028-1997 *IEEE Standard for Software Reviews* es otro manual para realizar revisiones en el software
- Identifica 37 productos software a revisar
- Un *producto software* es:
  - (i) Conjunto completo de programas, procedimientos, documentación y datos asociados
  - (ii) cada uno de los objetos individuales de (i)

# **Revisiones Técnicas Formales**

## **El IEEE Std. 1028-1997**

- Distingue entre cinco tipos de revisiones en función del producto a revisar:
  - Revisiones de gestión.
  - Revisiones técnicas.
  - Inspecciones.
  - Ensayos.
  - Auditorias.

# **Revisiones Técnicas Formales**

## **El IEEE Std. 1028-1997**

- Revisiones de gestión
  - Su propósito es:
    - Monitorizar el progreso
    - Determinar el estado de planes y planificaciones
    - Confirmar los requisitos y su localización en el sistema
    - Evaluar la efectividad de las técnicas de gestión para conseguir sus objetivos

# **Revisiones Técnicas Formales**

## **El IEEE Std. 1028-1997**

- Sirven para tomar decisiones acerca de:
  - Acciones correctivas
  - Cambios en la asignación de recursos
  - Cambios en el ámbito del proyecto
- Identifican:
  - Consistencia con y desviaciones de planes
  - Seguimiento, o falta del mismo de los procedimientos de los planes de gestión

# **Revisiones Técnicas Formales**

## **El IEEE Std. 1028-1997**

- Entre otros revisan:
  - Informes de auditoría
  - Planes de contingencia
  - Quejas del cliente
  - Informes de progreso
  - Plan del proyecto
  - Planes de gestión del riesgo
  - Planes SQA
  - Informes de revisiones técnicas



# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Revisiones técnicas
  - Su propósito es evaluar un producto software para:
    - Determinar su adecuación al uso pretendido
    - Identificar discrepancias con respecto a especificaciones y estándares
  - Sirven para determinar si:
    - El producto software se ajusta a sus especificaciones
    - El producto software se ajusta a sus regulaciones, estándares, guías, planes y procedimientos aplicables

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Los cambios en el producto software se implementan de manera adecuada y afectan únicamente a aquellas áreas del sistema identificadas por la especificación del cambio
- Entre otros revisan:
  - SRS
  - Diseño
  - Documentación de prueba
  - Documentación de usuario
  - Manual de mantenimiento
  - Procedimientos de instalación

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Inspecciones
  - Su propósito es detectar e identificar anomalías en el producto software
  - Son más detalladas que las revisiones técnicas
  - Es una evaluación por pares sistemática que sirve para:
    - Verificar que el producto software satisface sus especificaciones
    - Verificar que el producto software satisface los atributos de calidad especificados

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Verificar que el producto software se ajusta a las regulaciones aplicables, estándares, guías, planes y procedimientos
- Identificar desviaciones con respecto a estándares y especificaciones
- Opcionalmente, recolectar datos de IS (p.e. datos de anomalías o esfuerzo)
- Opcionalmente, utilizar los datos de IS recolectados para mejorar el proceso de inspección y su documentación de soporte (p.e. listas de comprobación)

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Durante las inspecciones:
  - Es obligatorio determinar las soluciones o investigar acciones para las anomalías, aunque la solución no se proponga durante la inspección
  - La recolección de datos de IS no es obligatoria, pero sí recomendable
  - Se puede optar por:
    - Aceptar el producto
    - Rechazar el producto
    - Aceptarlo con cambios

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Entre otros revisan:
  - SRS
  - Diseño
  - Código fuente
  - Documentación de prueba
  - Documentación de usuario
  - Manual de mantenimiento
  - Procedimientos de instalación
- Es la revisión más parecida a una RTF

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Ensayos
  - Su propósito es evaluar un producto software, aunque también podría utilizarse con fines formativos
  - Sirve para:
    - Encontrar anomalías
    - Mejorar el producto software
    - Considerar implementaciones alternativas
    - Evaluar el ajuste a estándares y especificaciones
    - Entrenamiento de los participantes

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

— Entre otros revisan:

- SRS
- Diseño
- Código fuente
- Documentación de prueba
- Documentación de usuario
- Manual de mantenimiento
- Procedimientos de instalación



# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

- Auditorías
  - Su propósito es proporcionar una evaluación independiente del ajuste de los productos y procesos software a:
    - Regulaciones
    - Estándares
    - Guías
    - Planes
    - Procedimientos

# Revisiones Técnicas Formales

## El IEEE Std. 1028-1997

— Entre otros revisan:

- Planes de contingencia
- Contratos
- Quejas de clientes
- Planes de mantenimiento
- Informes de revisiones de gestión
- Manuales de operación y de usuario
- Métodos de contratos
- Informes y datos (p.e. revisiones de auditorías, estado del proyecto, informes de anomalías)

# **Revisiones Técnicas Formales**

## **El IEEE Std. 1028-1997**

- Planes de gestión del riesgo
- Planes de gestión de la configuración software
- Descripciones de diseño
- Código fuente
- Planes de proyecto
- Planes de garantía de calidad
- SRS
- Documentación de prueba
- Documentación de usuario

# **Revisiones Técnicas Formales**

## **El IEEE Std. 1028-1997**

- Estándares, regulaciones, guías y procedimientos
- Informes de revisiones técnicas
- Informes de ensayos

# Garantía de calidad estadística

## Mejora estadística

- La *garantía de calidad estadística* de IS, no tiene nada que ver con la aplicación de técnicas estadísticas en el control de muestras
- Como ya hemos comentado en el apartado de métricas:
  - Se agrupa y clasifica la información sobre los fallos del software.
  - Se intenta encontrar la causa subyacente a cada fallo.
  - Se aplica el principio de Pareto.
  - Se actúa para corregir los fallos vitales.

# Garantía de calidad estadística

## Mejora estadística

Error	Total		Grave		Moderado		Leve	
	No.	%	No.	%	No.	%	No.	%
IEE	205	22%	34	27%	68	18%	103	24%
MCC	156	17%	12	9%	68	18%	76	17%
DDE	48	5%	1	1%	24	6%	23	5%
IEP	25	3%	0	0%	15	4%	10	2%
ERD	130	14%	26	20%	68	18%	36	8%
IMI	58	6%	9	7%	18	5%	31	7%
ELD	45	5%	14	11%	12	3%	19	4%
PIE	95	10%	12	9%	35	9%	48	11%
DII	36	4%	2	2%	20	5%	14	3%
TLP	60	6%	15	12%	19	5%	26	6%
IHM	28	3%	3	2%	17	4%	8	2%
VAR	56	6%	0	0%	15	4%	41	9%
Totales	942	100%	128	100%	379	100%	435	100%

#total: IEE, MCC y ERD

#graves: IEE, ERD, TLP y ELD

Recolección de datos  
para SQA estadística

# Garantía de calidad estadística

## Índice de errores

- El *índice de errores* permite cuantificar la magnitud de los fallos durante el proceso de desarrollo

# Garantía de calidad estadística

## Índice de errores

$$IE = (\sum_{i \in \text{fases}} iIF_i)/T$$

donde

T: tamaño del software (LDC, documentación, diseño)

$$IF_i = p_g g_i / e_i + p_m m_i / e_i + p_l l_i / e_i$$

donde

$g_i, m_i, l_i$ : # errores graves, moderados y leves en la etapa  $i$ -ésima

$e_i$ : # errores en la etapa  $i$ -ésima

$p_g, p_m, p_l$ : pesos grave (10), moderado (3) y leve (1)



# Garantía de calidad estadística

## Índice de errores

- Tal y como dice Pressman la SQA estadística se puede resumir en: “¡Utilizar el tiempo para centrarse en cosas que realmente interesan, pero primero asegurarse que se entiende qué es lo que realmente interesa!”

# Fiabilidad del software

- Ya comentado en el Capítulo 7 (t49):
  - Si de la construcción u operación del sistema se derivan riesgos para la salud, la SQA es primordial.
- En cualquier caso, siempre podemos medir la fiabilidad del software
- Fiabilidad del software: ausencia de fallos

# Fiabilidad del software

- *Probabilidad de Fallo Bajo Demanda (PFBD)*
  - Mide la probabilidad de que un sistema falle cuando se le hace una petición de servicio.
  - $PFBD = \frac{\text{\#fallos}}{\text{\#peticiones}}$
  - ↓mejor.
  - e.g. una PFBD de 0,001 significa que el sistema tiene un fallo cada mil peticiones.

# Fiabilidad del software

- *Frecuencia de fallo*
  - Mide la frecuencia de aparición de fallo de funcionamiento.
  - $FDF = \text{\#fallos} / \text{\#unidad tiempo}$
  - ↓mejor
  - e.g. una FDF de 0,006 indica que se producen 3 fallos cada 500 unidades de tiempo

# Fiabilidad del software

- Discusión: ¿es una probabilidad la FDF?
- *Tiempo medio de fallo* (TMF)
  - Mide el tiempo transcurrido entre fallos del sistema.
  - $\text{TMF} = \text{\#unidades de tiempo} / \text{\#fallos}$
  - $\uparrow$ mejor.
  - e.g. un TMF de 166,66 indica que transcurren 500 unidades de tiempo cada 3 fallos.
  - Si no hay cambios,  $\text{TMF} = 1/\text{FDF}$

# Fiabilidad del software

- *Disponibilidad*
  - Mide la disponibilidad de un sistema para ser usado.
  - $\text{dispon.} = \frac{\text{\# tiempo disponible}}{\text{\# tiempo funcionando}}$
  - $\uparrow$ mejor.
  - e.g. Una disponibilidad de 0,95 indica que el sistema está disponible 950 unidades de cada 1000 unidades de tiempo.
- Unidad de tiempo (CPU, días, etc.)

# **Plan SQA**

## **El IEEE Std. 730-2002**

- El IEEE Std. 730-2002 describe la preparación y contenidos de los planes SQA
- Un plan SQA sirve como guía de las actividades de SQA en el proyecto
- Lo desarrolla el equipo SQA

# **Plan SQA**

## **El IEEE Std. 730-2002**

- El índice del plan SQA es:
  1. Propósito
  2. Documentos de referencia
  3. Gestión
  4. Documentación
    - 4.1 Propósito
    - 4.2 Requisitos mínimos de documentación
    - 4.3 Otra documentación
  5. Estándares, prácticas, convenciones y métricas
    - 5.1 Propósito
    - 5.2 Contenido



# **Plan SQA**

## **El IEEE Std. 730-2002**

### 6. Revisiones del software

#### 6.1 Propósito

#### 6.2 Requisitos mínimos

#### 6.3 Otras revisiones y auditorias

### 7. Pruebas

### 8. Informe de errores y acciones correctoras

### 9. Herramientas, técnicas y metodologías

### 10. Control de medios

### 11. Control de proveedor

### 12. Colección de registros, mantenimiento y conservación

# **Plan SQA**

## **El IEEE Std. 730-2002**

13. Formación

14. Gestión del riesgo

15. Glosario

16. Procedimiento de cambio e historial del plan de SQA

- No toda la información listada tiene porque encontrarse en el documento

# **Plan SQA**

## **El IEEE Std. 730-2002**

- Propósito (1)
  - Delinea el propósito específico y el alcance del plan SQA.
  - Lista los nombres de los elementos software cubiertos por el plan SQA y el uso de dichos elementos.
  - Determina la porción del ciclo de vida cubierta por el plan para cada elemento software.

# Plan SQA

## El IEEE Std. 730-2002

- Documentos de referencia (2)
  - Proporciona una lista completa de cualquier documento referenciado en el plan o utilizado en su elaboración.
- Gestión (3)
  - Tareas relacionadas con la SQA.
  - Idealmente redactado en formato IEEE Std. 1058-1998, *IEEE Standard for Software Project Management Plans*

# **Plan SQA**

## **El IEEE Std. 730-2002**

- Describe la estructura organizativa que influye y controla la calidad del software.
- Identifica roles y responsabilidades dentro del plan SQA.
- Identifica a los responsables de preparar y mantener el plan SQA.
- Identifica las tareas asociadas al proceso SQA.
- Identifica la relación entre estas tareas y las de la planificación temporal

# Plan SQA

## El IEEE Std. 730-2002

- Documentación (4)
  - Define toda la documentación que se va a generar durante el proceso de desarrollo.
  - Lista los documentos que serán revisados o auditados, así como los criterios de revisión.
- Estándares, prácticas convenciones y métricas (5)
  - Esta sección es un poco *miscelánea en SQA*.

# **Plan SQA**

## **El IEEE Std. 730-2002**

- Revisiones del software (6)
  - Determina las revisiones del software.
  - Define los tipos de revisión.
  - Define los productos a revisar.
- Prueba (7)
  - Identifica todas las pruebas no incluidas en el plan de verificación y validación.

# **Plan SQA**

## **El IEEE Std. 730-2002**

- Informe de problemas y acción correctiva (8)
  - Describe las prácticas y procedimientos de informe, seguimiento y resolución de problemas, tanto a nivel producto como proceso.
  - Determina las responsabilidades organizativas relativas a su implementación.



# **Plan SQA**

## **El IEEE Std. 730-2002**

- Herramientas, técnicas y metodologías (9)
  - Herramientas, técnicas y metodologías utilizadas para soportar el proceso de SQA.
- Control de medios (10)
  - Determina los métodos para:
    - Identificar el medio físico de cada producto software.
    - Protegerlo de daños durante el proceso.

# **Plan SQA**

## **El IEEE Std. 730-2002**

- Control de proveedor (11)
  - Determina las técnicas para garantizar que el software proporcionado por proveedores externos cumple sus requisitos.
  - También es aplicable a código heredado.

# **Plan SQA**

## **El IEEE Std. 730-2002**

- Colección de registros, mantenimiento y conservación (12)
  - Identifica la documentación SQA que no se debe tirar tras acabar el proceso.
  - Determina los métodos y medios para ensamblar, archivar, salvaguardar y mantener la documentación.
  - Fija el periodo de conservación de la información.

# Plan SQA

## El IEEE Std. 730-2002

- Formación (13)
  - Identifica las actividades de formación necesarias para satisfacer las necesidades del plan SQA.
- Gestión del riesgo (14)
  - Especifica el plan de gestión del riesgo.
  - Idealmente redactado según IEEE Std. 1540-2001 *for Software Life Cycle Processes – Risk Management*

# **Plan SQA**

## **El IEEE Std. 730-2002**

- **Glosario (15)**
  - Términos específicos del plan SQA.
- **Procedimiento de cambio e historia del plan SQA (16)**
  - Procedimientos de modificación del plan SQA.
  - Procedimientos de mantenimiento del historial de cambios.
  - Historial de cambios.

# El enfoque de calidad ISO 9000

## Introducción

- La *International Organization for Standarization* ISO\* tiene el enfoque ISO 9000 para la gestión de la calidad
- El enfoque ISO 9000 busca la gestión a través de la implantación de un *sistema de gestión de la calidad* QMS (*Quality Management Systems*)

\*<http://www.iso.org>

# El enfoque de calidad ISO 9000

## Introducción

- ISO 9000 es una familia de estándares *genéricos*
- De esta forma pueden aplicarse a distintos sectores productivos (software, hardware, automoción, textil, etc.)

# El enfoque de calidad ISO 9000

## Introducción

- ISO 9000 es:
  - Una herramienta para clientes que *compran*, más que para fabricantes que *construyen*.
  - Se centra en *qué*, no en *cómo*. El estándar define un conjunto de objetivos a cumplir, pero no dice como conseguirlo.
  - Define elementos *necesarios*, pero no *suficientes*.



# El enfoque de calidad ISO 9000

## La familia ISO 9000

- ISO 9000:2005 *Quality management systems - Fundamentals and vocabulary*
  - Determina un punto de partida para comprender la familia de estándares y define los términos y definiciones fundamentales utilizados en la familia ISO 9000

# El enfoque de calidad ISO 9000

## La familia ISO 9000

- ISO 9001:2000 *Quality management systems - Requirements.*
  - Identifica los requisitos que una organización debe satisfacer para tener implementado un sistema de gestión de calidad ISO 9000.
- ISO 9004:2000 *Quality management systems – Guidelines for performance improvement*
  - Guía para una continua mejora del proceso.

# El enfoque de calidad ISO 9000

## La familia ISO 9000

- ISO/IEC 90003:2004 *Software engineering- Guidelines for the application of ISO 9001:2000 to computer software*
  - Es una guía de aplicación del estándar ISO 9001:2000 en organizaciones que se dediquen al desarrollo de software.
    - Norma española UNE-ISO/IEC 90003 de AENOR (64 páginas)

# **El enfoque de calidad ISO 9000**

## **Relación con otros estándares**

- El modelo de capacidad de madurez SEI
  - ISO 9001 es genérico, y CMM específico para la industria del software.
  - CMM es más detallado y específico.
  - ISO 9001 es binario, mientras que CMM permite cinco niveles.
  - ISO 9001 se centra en la relación cliente-proveedor, mientras que CMM se centra en el desarrollo de software.

# **El enfoque de calidad ISO 9000**

## **Relación con otros estándares**

- IEEE Std. 730
  - IEEE Std. 730 se centra en el plan de garantía de calidad, y en como implementarlo, mientras que ISO 9001 busca una gestión global de la calidad centrada en auditorias externas.

# El enfoque de calidad ISO 9000

## AENOR

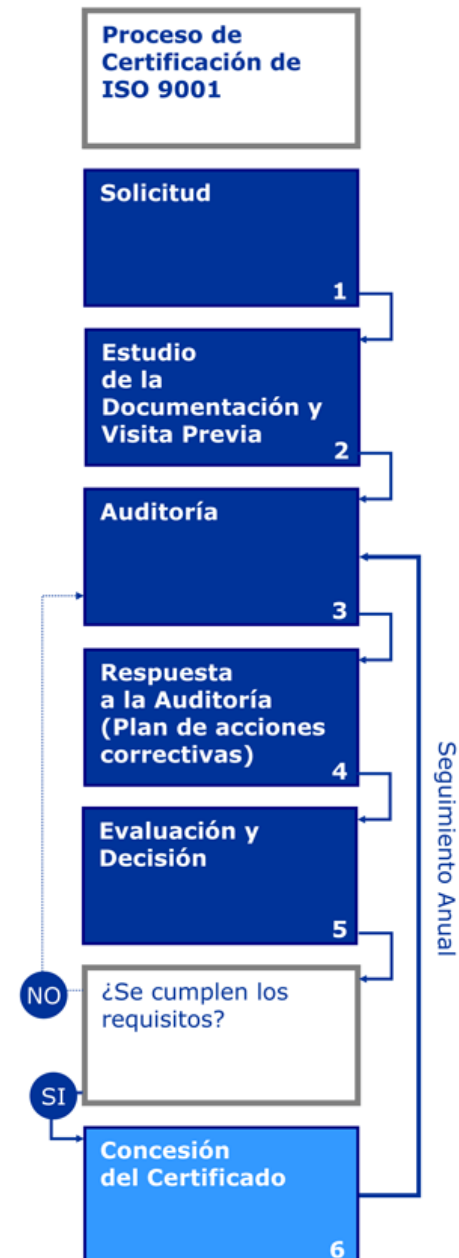
- AENOR\*
  - Entidad privada, española, independiente y sin ánimo de lucro.
  - Dedicada a:
    - Elaborar normas técnicas españolas.
    - Certificar productos, servicios y empresas.
  - *En particular* proporciona certificaciones ISO 9000.

\*<http://www.aenor.es>

# El enfoque...

## AENOR

### Proceso de certificación ISO 9001 de AENOR



# Conclusiones

- Calidad: base de IS
- Calidad de diseño vs. calidad concordancia
- Variación entre muestras
- SQA
- Calidad: concordancia
- Control de calidad y garantía de calidad
- Coste de calidad: asumible



# Conclusiones

- Equipo SQA
- V&V. IEEE Std. 1012-2004
- RTFs
- IEEE Std. 1028-1997
- MEPS
- Plan SQA – IEEE Std. 730-2002
- ISO 9000
- *Sentido común*