



# Tema 10 – Garantía de calidad del software

## Ingeniería del Software

Héctor Gómez Gauchía  
Dep. Ingeniería del Software e Inteligencia Artificial  
Facultad de Informática  
Universidad Complutense Madrid

Trabajando con Rubén Fuentes, Antonio Navarro, Juan Pavón y Pablo Gervás





# Contenidos

- Introducción
  - Problemática
- Conceptos de calidad
- Verificación y Validación
  - IEEE Std. 1012-2004
- Revisiones del software
  - IEEE Std. 1028-2008
- Garantía de calidad del software
- Fiabilidad del software
- Plan de Aseguramiento de la Calidad del Software
- ISO 9000





# Introducción

- Uno de los objetivos principales de la Ingeniería del Software (IS) es construir software de calidad.
- ¿Cómo medimos la calidad del software?
  - Con medidas indirectas.
    - No hay un atributo que represente la *calidad*.
  - Podemos definir calidad en base a características o atributos de elementos del producto o proceso.
- Dificultad clave:
  - En ingeniería, el aseguramiento de calidad se centra en el control de variación entre muestras.
  - En IS, la variación entre muestras no tiene sentido.
    - Cada proyecto es único.





# 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
- El IEEE Std. 610.12 [IEEE C/S2ESC *Software & Systems Engineering Standards Committee*, 1990] 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?*





# Tipos de calidad

- Calidad de diseño
  - Se basa en las características especificadas para un elemento.
  - Típicamente a través de la Especificación de Requisitos Software (*Software Requirements Specification, SRS*) y el Diseño.
- Calidad de concordancia
  - Se centra en el grado de cumplimiento de las especificaciones de diseño durante su realización.
  - Se mide en la implementación del software.





# Obtención de calidad

- 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 (*Software Quality Assurance, SQA*).





# 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.
- 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.
- Incluyen bucles de realimentación.





# Aspectos de la SQA

- 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







# Actividades del grupo SQA

- Establecer el plan SQA para el proyecto
- Participar en el desarrollo de la descripción del proceso de software
- Revisar las actividades de IS para verificar su ajuste al proceso del software
- Auditar los productos de software designados para verificar el ajuste con los definidos como parte del proceso de software
- 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





# 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 con la obtención de la calidad.
- Hay tres tipos de costes de calidad:
  - De prevención
  - De evaluación
  - De fallos





# Tipos de costes de calidad

- Costes de prevención
  - Planificación de calidad
  - Revisiones Técnicas Formales (RTFs)
  - Equipo de pruebas
  - Formación
- Costes de evaluación
  - Inspección en el proceso y entre procesos
  - Calibrado y mantenimiento del equipo
  - Pruebas
- Costes de fallos
  - Internos
  - Externos





# Fallos internos

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





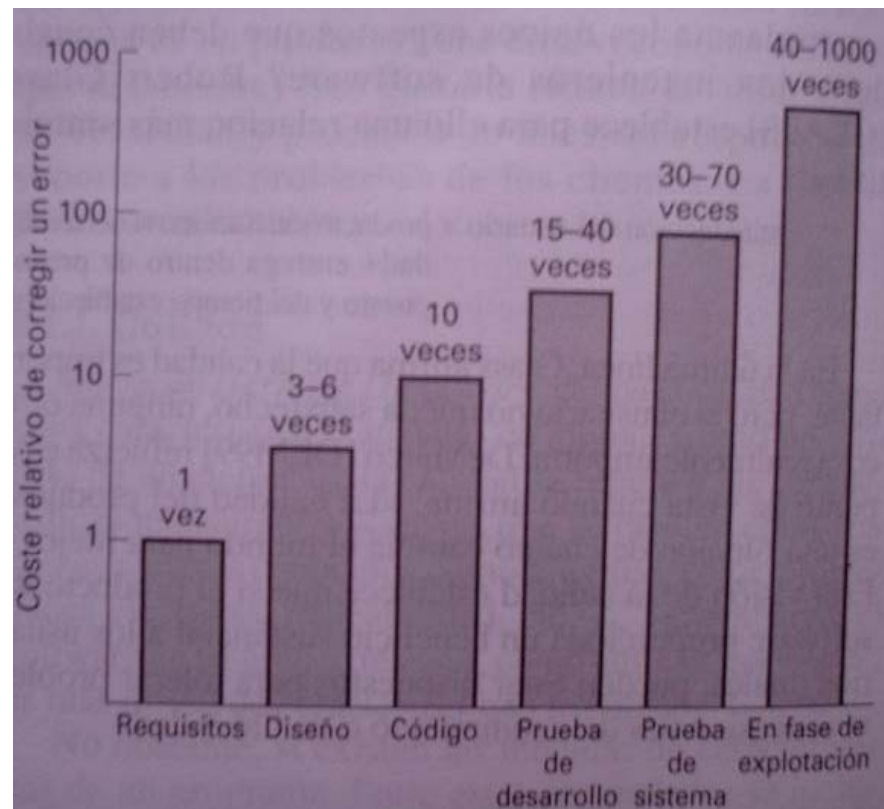
# Fallos externos

- 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



## Coste del retraso

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

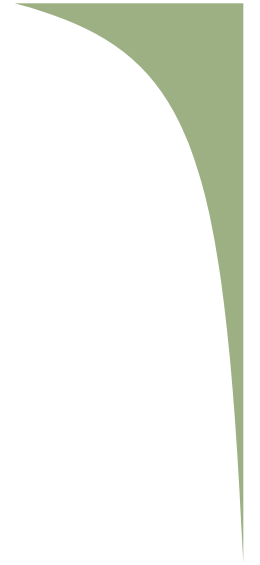




## Ejemplo: coste del retraso

- Un análisis del trabajo de IBM en Rochester indica:
  - 7053 horas de RTFs de 200000 LDC (Líneas De Código)
  - 3112 errores descubiertos
  - Coste programador 40 \$/h
  - Coste eliminación defectos: 282120 \$ (91 \$/defecto)
- Supongamos que en vez de RTFs somos muy cuidadosos.
  - Supongamos que hay 1 defecto por 1000 LDC.
  - 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.





# VERIFICACIÓN Y VALIDACIÓN







# Introducción

- Verificación y Validación (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.
- 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.





## IEEE Std. 1012-2004

- El IEEE Std. 1012-2004 [IEEE 1012 WG, 2004] define los procesos de V&V en términos de actividades específicas y tareas relacionadas.
  - También define los contenidos del plan de V&V del software, incluyendo un ejemplo de formato.
- Identifica 6 procesos a los cuales se puede aplicar V&V:
  - Gestión
  - Adquisición
  - Suministro
  - Desarrollo
  - Operación
  - Mantenimiento

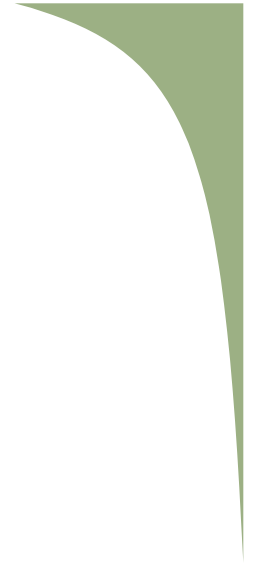




# IEEE Std. 1012-2004: niveles de integridad

- Un nivel de integridad es un conjunto de 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.
- El estándar identifica 4 niveles de integridad.
  - Nivel 4 → Consecuencias graves e irreparables en caso de fallo.
  - Nivel 3 → Consecuencias serias en caso de fallo.
  - Nivel 2 → Consecuencias menores en caso de fallo.
  - Nivel 1 → Consecuencias insignificantes en caso de fallo.





# REVISIONES DEL SOFTWARE



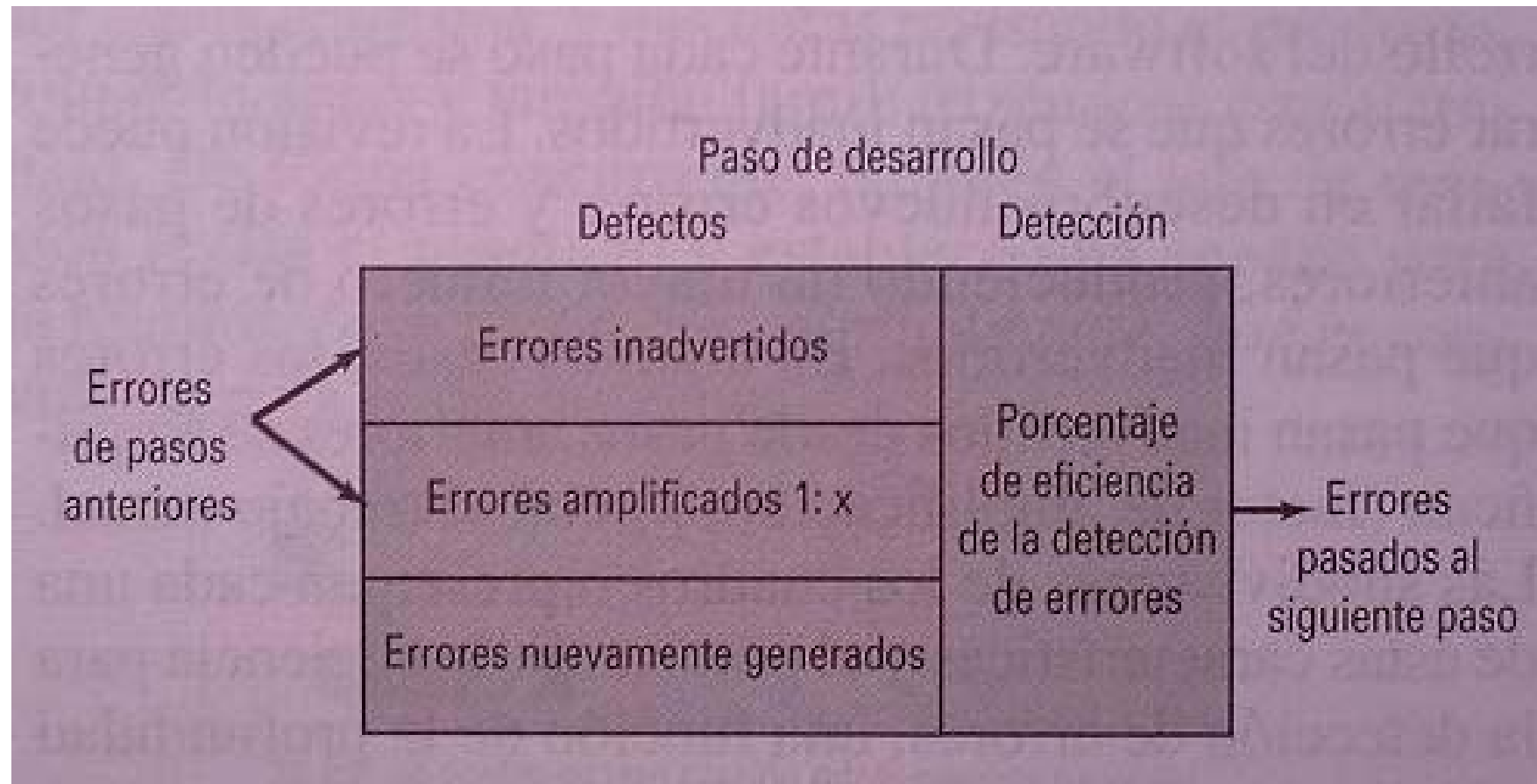


# Introducción

- 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.
  - Aquí 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.



# Modelo de amplificación de defectos



- Una RTF [*Office of Safety and Mission Assurance, NASA, 1993*] 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
  - Conseguir un software desarrollado de manera uniforme
  - Hacer que los proyectos sean más manejables



# Reunión de revisión: principios básicos

- La RTF se instrumenta mediante una reunión, que debe ser convenientemente planificada.
- Normas de la reunión:
  - Deben convocarse para la reunión 3-5 personas.
  - Se debe preparar por adelantado, pero sin que requiera más de 2 horas de trabajo por persona.
  - La duración debe ser menor de 2 horas.
- Evidentemente, la RTF se centra en partes específicas del software.
  - Se deben revisar los productos resultantes de las tareas de trabajo.







# Reunión de revisión: previo

- 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 2-3 revisores.
  - Cada revisor, incluido el jefe de revisión, revisa el producto durante 1-2 horas.
  - El jefe de revisión establece una agenda y planifica una reunión.
    - Normalmente para el día siguiente.





# Reunión de revisión: realización

- El procedimiento es:
  - A la reunión asisten el productor, los revisores y el jefe de revisión.
  - Uno de los revisores juega el papel de registrador.
  - El productor expone su producto.
  - Los revisores exponen sus problemas o reservas.
  - Cuando se descubre un problema o error real, el registrador toma nota.





# Reunión de revisión: cierre

- El procedimiento es:
  - 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.
      - Asignar encargado del seguimiento de las modificaciones
  - La reunión se cierra con la firma del registro de revisión por los participantes.
    - Se indican los asistentes y su nivel de conformidad.





# Reunión de revisión: 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 todos los problemas.
  - Identifica las áreas problemáticas dentro de un producto.
  - Sirve como lista de comprobación de puntos de acción que guíe al productor para hacer las correcciones.
- Además se genera un informe sumario de la revisión.
  - Qué se revisó.
  - Quién lo revisó.
  - Qué se descubrió, y cuáles son las conclusiones.
- El informe es una página simple que se adjunta al registro histórico del proyecto.
  - Pueden añadirse los anexos que se consideren necesarios.





# Directrices de revisión

- 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 para cada producto que vaya a ser revisado.
- Disponer de recursos y una agenda para la RTF.
- Entrenar a todos los revisores.
- Repasar las reuniones anteriores.

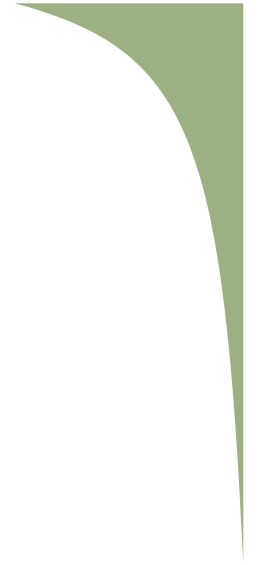




## Recomendaciones para RTFs

- El “*Software Formal Inspections Guidebook*, NASA-GB-A302” [Office of Safety and Mission Assurance, NASA, 1993] incluye listas de comprobación de productos software.
- El IEEE Std. 1028-2008 [IEEE 1028 WG, 2008] identifica 37 productos software a revisar y da pautas para hacerlo.
  - Revisiones de gestión
  - Revisiones técnicas
  - Inspecciones
  - Ensayos
  - Auditorias





# GARANTÍA DE CALIDAD ESTADÍSTICA





# Introducción

- 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 tema 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.





# Índice de errores

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

$$IE = \frac{\sum_{i \in fases} (i * IF_i)}{T}$$

- donde:
  - T es el tamaño del software (LDC, documentación, diseño...)

$$IF_i = \frac{p_g * g_i + p_m * m_i + p_l * l_i}{e_i}$$

- donde
  - $g_i$ ,  $m_i$  y  $l_i$  son el número de errores graves, moderados y leves en la etapa i-ésima respectivamente.
  - $e_i$  es el número de errores en la etapa i-ésima.
  - $p_g$ ,  $p_m$  y  $p_l$  son los pesos de los errores graves (10), moderados (3) y leves (1) respectivamente.





# Fiabilidad del software

- 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.
  - Probabilidad de Fallo Bajo Demanda (PFBD)
    - $PFBD = \text{\#fallos} / \text{\#peticiones}$
  - Frecuencia de Fallo (FF)
    - $FF = \text{\#fallos} / \text{\#unidades de tiempo}$
  - Tiempo Medio de Fallo (TMF)
    - $TMF = \text{\#unidades de tiempo} / \text{\#fallos}$
  - Disponibilidad
    - $\text{Disponibilidad} = \text{\#tiempo disponible} / \text{\#tiempo funcionando}$
- Fiabilidad del software = ausencia de fallos





# PLAN DE SQA





## IEEE Std. 730-2002

- El IEEE Std. 730-2002 [IEEE 730 WG, 2002] describe la preparación y contenidos de los planes SQA.
  - No toda la información listada tiene porque encontrarse en el documento.
- Un plan SQA sirve como guía de las actividades de SQA en el proyecto.
- Lo desarrolla el equipo SQA.





## IEEE Std. 730-2002: índice del plan de SQA (1/2)

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
6. Revisiones del software
  - 6.1 Propósito





## IEEE Std. 730-2002: índice del plan de SQA (2/2)

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

13. Formación

14. Gestión del riesgo

15. Glosario

16. Procedimiento de cambio e historial del plan de SQA





# ISO 9000





# ISO 9000

- La Organización Internacional para la Estandarización (ISO) tiene el enfoque ISO 9000 para la gestión de la calidad.
  - ISO 9000 es una familia de estándares genéricos.
  - De esta forma pueden aplicarse a distintos sectores productivos.
    - Ej. software, hardware, automoción o textil.
  - <http://www.iso.org>
- El enfoque ISO 9000 busca la gestión a través de la implantación de un Sistema de Gestión de la Calidad (*Quality Management System*, QMS).







# ISO 9000: conceptos esenciales

- Calidad
  - Grado en el que un conjunto de características cumplen sus requisitos.
- Gestión de la calidad
  - Actividades coordinadas para dirigir y controlar una organización en lo concerniente a la calidad.
- Sistema de gestión de calidad
  - Conjunto de directrices e instrucciones cuyo objetivo es alcanzar unos niveles satisfactorios y constantes de calidad en productos y/o servicios.





# ISO 9000: perspectiva y principios

- 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.





## La familia ISO 9000 (1/2)

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





## La familia ISO 9000 (2/2)

- ISO 9004:2000, *Quality management systems – Guidelines for performance improvement*
  - Guía para una mejora continua del proceso.
- 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.

IO6



## Diapositiva 44

---

**IO6**

No es una errata.

IO; 10/01/2012



# Guías sobre aspectos específicos del QMS

- ISO 10006 para gestión del proyecto
- ISO 10007 para gestión de la configuración
- ISO 10012 para sistemas de medidas
- ISO 10013 para documentación de calidad
- ISO/TR 10014 para gestionar la economía de la calidad
- ISO 10015 para formación
- ISO/TS 16949 para suministradores de automoción
- ISO 19011 para auditoría





# Implementación ISO 9001:2000

1. Identificar los objetivos a alcanzar.
2. Identificar qué se espera por parte de los demás.
3. Obtener información acerca de la familia ISO 9000.
4. Aplicar la familia ISO 9000 de estándares al sistema de gestión.
5. Obtener guías sobre los aspectos específicos dentro del sistema de gestión de calidad.
6. Establecer el estado actual, y determinar las discrepancias entre el sistema de gestión de la calidad actual y las exigencias de ISO 9001:2000.
7. Determinar los procesos necesarios para suministrar productos a los clientes.
8. Desarrollar un plan para eliminar las discrepancias del paso 6 y desarrollar los procesos del paso 7.
9. Ejecutar el plan.
10. Realizar la evaluación periódica interna.
11. Si es necesario demostrar conformidad pasar a 12, si no pasar a 13.
12. Realizar auditorías externas.
13. Continuar con la mejora del negocio.





# Relación de ISO 9000 y el CMM

- El CMM es el Modelo de Madurez de las Capacidades del software (*Capability Maturity Model*, CMM) del Instituto de Ingeniería del Software de la Universidad de Carnegie Mellon (*Software Engineering Institute, Carnegie Mellon University*, SEI-CMU).
- Comparación:
  - 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.







## Relación de ISO 9000 con IEEE Std. 730

- EL IEEE Std. 730-2002 [IEEE 730 WG, 2002] es el estándar del IEEE que describe la preparación y contenidos de los planes SQA.
- Comparación:
  - IEEE Std. 730 se centra en el plan de garantía de calidad, y en como implementarlo.
  - ISO 9001 busca una gestión global de la calidad centrada en auditorias externas.





# CONCLUSIONES





# Conclusiones

- La búsqueda de la calidad en los productos software se encuentra en la base de la IS.
  - Como en todas las ingenierías.
- Existen diferentes perspectivas sobre la calidad.
  - En su definición, calidad de diseño vs calidad de concordancia
  - En cómo obtenerla, estándares de IEEE e ISO y recomendaciones NASA
- Los costes de la garantía de calidad han de ser asumibles.
  - Beneficios obtenidos vs coste del sistema de garantía de calidad
- La certificación de la calidad implica la evaluación por un organismo externo reconocido a tal efecto.
  - En España, AENOR.





## Glosario (1/2)

- AENOR = Asociación Española de NORmalización y certificación
- CMM = *Capability Maturity Model*
- FF = Frecuencia de Fallo
- IE = Índice de Errores
- IEC = *International Electrotechnical Commission*
- IEEE = *Institute of Electrical and Electronics Engineers*
- IS = Ingeniería del Software
- ISO = Organización Internacional para la Estandarización
- LDC = Líneas de Código
- NASA = *National Aeronautics and Space Administration, EEUU*





## Glosario (2/2)

- PFBD = Probabilidad de Fallo Bajo Demanda
- QMS = *Quality Management System*
- RTF = Revisión Técnica Formal
- SEI-CMU = *Software Engineering Institute, Carnegie Mellon - University*
- SQA = *Software Quality Assurance*
- SRS = *Software Requirements Specification*
- TMF = Tiempo Medio de Fallo
- UNE = Una Norma Española
- V&V = Verificación y Validación
- WG = *Working Group*





## Referencias (1/3)

- R. Pressman: Ingeniería del Software. Un enfoque práctico, 7ª edición. McGraw-Hill, 2010.
  - Capítulos 14-16
- I. Sommerville: Ingeniería del Software, 7ª edición. Addison Wesley, 2007.
  - Capítulo 27
- IEEE 1012 WG – Std. for Software Verification and Validation Working Group : IEEE Std. 1012-2004, IEEE Standard for Software Verification and Validation. IEEE, 2004.





## Referencias (2/3)

- IEEE 1028 WG – Std. for Software Reviews Working Group: IEEE Std. 1028-2008, IEEE Standard for Software Reviews. IEEE, 2008.
- IEEE 730 WG – Std. for Software Quality Assurance Plans Working Group: IEEE Std. 730-2002, IEEE Standard for Software Quality Assurance Plans. IEEE, 2002.
- IEEE C/S2ESC Software & Systems Engineering Standards Committee: IEEE Std. 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology. IEEE, 1990.





## Referencias (3/3)

- ISO: ISO 9001:2000, Quality management systems – Requirements. ISO, 2000.
- ISO: ISO 9004:2000, Quality management systems – Guidelines for performance improvement. ISO 2000.
- ISO/IEC: ISO/IEC 90003:2004, Software engineering - Guidelines for the application of ISO 9001:2000 to computer software. ISO, 2004
- ISO: ISO 9000:2005, Quality management systems - Fundamentals and vocabulary. ISO, 2005.
- Office of Safety and Mission Assurance, NASA: Software Formal Inspections Guidebook. NASA-GB-A302, NASA, 1993.

