
Planificación y Gestión de Proyectos Informáticos

Tema 9. Gestión de la Calidad en un proyecto informático

**Máster en Ingeniería Informática
Universidad de Granada**

Objetivos

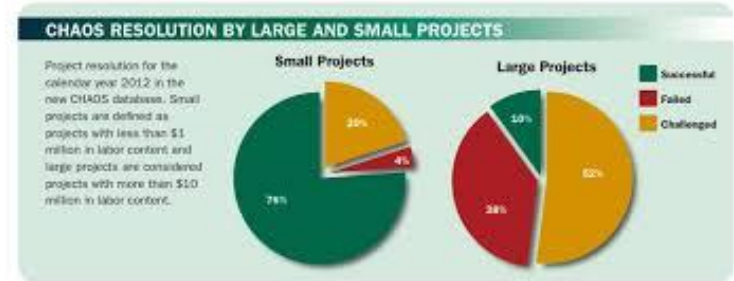
- Descubrir qué son la calidad del producto y del proceso, y que están relacionadas.
- Aprender a planificar y establecer mejoras del proceso.
- Conocer algunos estándares y certificaciones de calidad.
- Concretar tareas relacionadas con la garantía de la calidad, a realizar durante todo el ciclo de vida del software.

Contenidos

1. Definición de Calidad
2. Planificación de la Calidad
3. Mejora de la Calidad
4. Estándares y Certificaciones
5. Gestión de la calidad: concretando qué hacer

1. Definición de calidad

[Pantaleo, 2012]



Un **producto es de calidad** si le sirve a quien lo adquiere y éste lo usa para realizar las tareas para las que fue concebido, cumpliendo unos requisitos prefijados.

Un **proceso es de calidad** si genera productos de buena calidad, para lo cual debe estar bien concebido y bien implementado.

2. Planificación para la calidad

- **Recursos humanos:** contar con personal motivado, formado y especializado, capacitado en estándares y herramientas. Involucrar a clientes.
- **Infraestructura:** equipamiento actualizado, potente y diverso. Elección de arquitectura de base hardware y software, garantía de personal de mantenimiento y soporte técnico.
- **Metodología:** Proceso planificado (realista) y basado en estándares. Evaluación de objetivos y resultados, y cumplimiento de compromisos.

Mejores resultados: Agiles, TDD (pruebas)

3. Mejora de calidad

Implica **reconocimiento de necesidad de la mejora** respecto a lo actual.

Medidas cuantitativas de lo actual: productividad, el esfuerzo, los defectos detectados, riesgos no previstos, etc.

Requiere **definir y documentar** por escrito los procesos a mejorar, **formando** a los que los van a utilizar y asignándoles **responsabilidades**.

→ **inconveniente:** resistencia al cambio

Resistencia al cambio

La mejora implica un cambio que no todos quieren hacer.

Motivos de la resistencia al cambio:

Diferencia de opiniones, incertidumbre respecto a roles y tareas, pérdida de estatus, riesgo mayor que continuar como se está.

En muchos casos, para decidirse, se requiere una **estimación** de los costes que supondrán la realización del cambio (modelos de estimación).

Plan de mejora de proceso: ejemplo

Mejora que se quiere: Reducir el número de incidencias (quejas) de los usuarios durante el mantenimiento.

Tareas para mejorar: Crear un equipo de inspección y realizar más revisiones antes de la entrega.

Hay que valorar si la decisión es correcta, basada en métricas

Qué medir:

1. Quejas de clientes por unidad de tiempo y tamaño de software.
2. Esfuerzo dedicado a revisiones por unidad de tamaño del producto.
3. Personal disponible para pruebas y mantenimiento

Valoración: Sí o no, según las medidas tomadas.

4. Estándares de Calidad

Establecen **cómo revisar y adaptar los procesos de software**, y cómo incorporar nuevos procesos que ayuden a **generar productos de mejor calidad**.

Implican **medidas de seguimiento y métricas de valoración** cuantitativas y cualitativas, con el objetivo de validar que ha habido mejoras.

El **seguimiento de estándares** no debe incrementar los costes ni retrasar los proyectos por su aplicación.



Certificaciones de Calidad



Las certificaciones de calidad indican que los procedimientos para construir un software son correctos y están en función de estándares de calidad.



Se certifica la empresa y la metodología para el desarrollo.

La certificación implica una auditoría interna y una externa, emitida por un organismo autorizado. Si los resultados de una certificación son positivos, se emite la certificación o acreditación y cada cierto tiempo se tiene que renovar.

Ejemplos guías y estándares de calidad

MADEJA: Marco de Desarrollo de la Junta de Andalucía.
Guías aplicables durante el ciclo de vida

ITIL: guía de buenas prácticas y recomendaciones para la administración de servicios de TI.

COBIT: framework con buenas prácticas para lograr sus objetivos de gobernanza y gestión de TI en empresas.

CMMI : modelo de madurez del proceso de desarrollo

ISO/IEC 20000: procesos de desarrollo y gestión de servicio.

ISO 9000 y ISO/IEC 25000: requisitos y calidad de un producto

Norma 27000 : seguridad de la información

Ejemplo Guía de CMMI

Ejemplos de buenas prácticas para satisfacer el nivel 2 de CMMI en un proceso de especificación de requisitos (se indica qué hacer, no cómo):

- Hacer modelado de requisitos
- Gestionar y administrar recursos para cubrir los requisitos
- Capacitar a los miembros del equipo en tareas de especificación y análisis de requisitos
- Garantizar la comunicación entre todos los stakeholders
- Evaluar la consistencia del trabajo realizado
- Utilizar métricas para evaluar el trabajo planificado
- Analizar las medidas tomadas, realizando informes
- Analizar los informes para planificar e implementar acciones correctivas

ISO/IEC 20000

Norma basada en procesos, no para evaluar productos. Ayuda para desarrollar herramientas, productos y sistemas que soporten las mejores prácticas de desarrollo y gestión del servicio.

Basada en la metodología **PDCA** (Plan-Do-Check-Act):

- **Planificar**: objetivos y procesos necesarios para proporcionar resultados de acuerdo con las necesidades del cliente y políticas de la empresa
- **Hacer**: implementar los procesos
- **Verificar**: monitorizar y medir los procesos
- **Actuar**: emprender acciones para mejorar continuamente el rendimiento y el comportamiento del proceso.

Ejemplo ISO/IEC 20000

Ejemplo de lo que dice la norma sobre **Information security management**

- a) Information security management objectives and plans to implement the security policy and the controls are agreed.
- b) Information security requirements have been communicated to all appropriate personnel in the service provider, customers and suppliers.
- c) The criteria for assessment of information security risks and the acceptable level of risk has been agreed.
- d) Regular information security risk assessments are conducted.
- e) Internal information security audits are conducted, the results are reviewed and any identified improvements recorded, agreed and implemented.
- f) Appropriate information security controls that have been implemented and operated to manage risks related to information security, meet the requirements of the information security policy.
- g) Information security controls are documented and describe the risks to which the controls relate, their operation and maintenance of the controls.
- h) All necessary information security requirements have been documented and agreed with external organizations that need to access, utilise or manage the service provider's information and services in order to reduce risks which can be caused by external parties.
- i) Information security controls are in place to enable the types, volumes and impacts of information security incidents to be quantified, reported and improvements identified.
- j) Requests for change are assessed to identify new or changed information security risks and for impact on existing information security controls.
- k) Information security incidents are managed through the incident management and request fulfilment process

Ejemplo ISO/IEC 9126

Norma para especificar la **calidad de un producto** (dentro de 9000).

<http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0032555&PDF=Si#.VHCO68kbi1s>

Calidad interna y externa: Orientada a evaluar el cumplimiento las siguientes características en los productos software (RNF):

- **Funcionalidad:** Adecuación, Exactitud, Interoperabilidad, Seguridad, Cumplimiento funcional.
- **Fiabilidad:** Madurez, Recuperabilidad, Tolerancia a fallos, Facilidad de recuperacion.
- **Usabilidad:** Facilidad de Aprendizaje y Comprensión, Operatividad, Atractividad
- **Eficiencia:** Comportamiento en el tiempo, Comportamiento de recursos,
- **Mantenibilidad:** Estabilidad, Facilidad de Análisis y Diagnóstico, Facilidad de Cambio, Facilidad de Pruebas
- **Portabilidad:** Facilidad de Instalación y Ajuste, y Adaptabilidad.

Calidad en el uso en un ambiente concreto por parte del usuario: efectividad, productividad, integridad, y satisfacción,

No establece métricas concretas ni prácticas para garantizarlas.

5. Gestión de la calidad: concretando qué hacer



Planificar acciones de:

- Prevención
- Evaluación
- Solución de fallos

Considerar el momento

Gestión de la calidad: concretando qué hacer

- Seguir metodologías de desarrollo estandarizadas y usar métodos y herramientas para especificación, diseño y codificación. Mantener un personal formado y actualizado.
- Controlar la documentación del software y de los cambios realizados. Garantizar la comunicación efectiva.
- Planificar y realizar revisiones técnicas formales (reuniones de trabajo para evaluar) y auditorías en cada fase.
- Aplicar estrategias de prueba: planificar y realizar pruebas sistemáticas.
- Aplicar métricas para medir aspectos que influyan en la calidad.
- Realizar informes sobre los cambios, resultado de revisiones, auditorías del software, pruebas,... y hacerlos públicos.

Revisiones técnicas formales

Técnica para gestión de calidad, con el propósito de:

Detectar errores y encontrar defectos en fases tempranas
Evitar propagación de errores a lo largo del desarrollo

Realizadas por los responsables de un proyecto, son **reuniones de trabajo periódicas**, que se deben preparar por adelantado y en las que **se estudia un proyecto y se toman decisiones**. ¿Qué se hace?:

- Listas de chequeo para revisar un producto o documento concreto.
- No se revisa al autor.
- Se fija un programa y lista de chequeo cerrada para la reunión, evitando discusiones interminables.
- Se detectan los problemas y no se resuelven.
- Se entrena a los revisores, limitando su número y seleccionándolos.
- Se hace un acta de las reuniones.

Auditorías

Evaluar que los objetivos del proyecto y de la organización se satisfacen.
Puede haber auditorías internas y externas, realizadas por personal certificado.
Se usan listas de chequeo, entrevistas, observación y herramientas específicas.

¿Cuándo hacer una auditoría?

- Inicio: Comprobar si el proyecto está bien planteado
- Durante el Funcionamiento: Comprobar conformidad con los procedimientos, normativas y planificación
- Cierre: Conformidad con los documentos contractuales y normativas de calidad

Etapas de una auditoría

- Estudio inicial y motivación para la auditoría
- Elaborar un plan de trabajo
- Desarrollar la auditoría
- Elaborar un diagnóstico y un informe final
- Elaborar un plan de ayuda, recomendando acciones correctoras
- Comprobar que se realizan las acciones correctoras y su utilidad

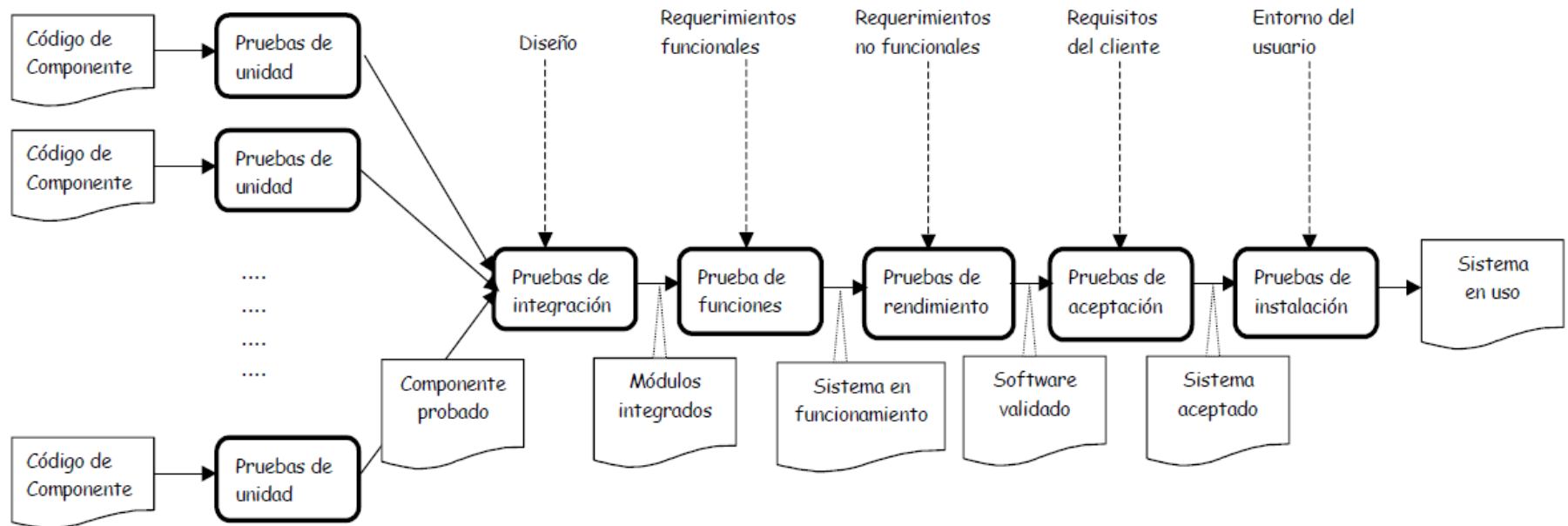
Pruebas

Su objetivo es descubrir fallos.

TIPOS DE FALLOS (no solo del software)

- **Algorítmico:** Fallo en la secuencia de procesamiento. Causas típicas incluyen: comprobar una condición incorrecta, no inicializar, no comprobar casos especiales, ramificar demasiado pronto o demasiado tarde.
- **Precisión:** Utilización de formulas incorrectas, errores de truncamiento o redondeo, o falta de precisión en los resultados.
- **Documentación:** La documentación no concuerda con el software.
- **Capacidad:** La respuesta del sistema es inaceptable cuando éste se carga.
- **Sincronización:** Error en momento datos de sincronización entre procesos.
- **Eficiencia:** El sistema no actúa a la velocidad requerida.
- **Recuperación:** El sistema no se comporta correctamente tras detectar un error.
- **Estándares y procedimientos:** El código no sigue las normas de codificación.

Pruebas



Sistema en uso: repetir pruebas para nuevas versiones, pruebas de regresión

Verificación: ¿Estamos construyendo el producto correcto? $2+3=5$ pero ¿queríamos sumar?

Validación: ¿Funciona el producto correctamente? $2*3=8$ ó $2*3=.....6$

Métricas de calidad, ejemplos

- Métricas para medir la **fiabilidad**:

- TIEMPO MEDIO ENTRE FALLOS :

$TMEF = TMDF + TMDR$ (Tiempo medio de fallo + tiempo medio de reparación)

- DISPONIBILIDAD:

Probabilidad de que un programa funcione de acuerdo con los requisitos en un momento dado: $Disponibilidad = TMDF / (TMDF + TMDR) \%$

- Métricas para valorar un buen **diseño** observando un código escrito en un lenguaje orientado a **objetos**:
 - Número de clases raíz en un programa (1).
 - Número de superclases de una clase (1).
 - Número de clases hijas de una clase (bajo).
 - Nivel de profundidad del árbol de jerarquía de herencia (bajo).

Tareas de este tema

1. Buscar en un estándar de calidad relacionado con tecnologías de la información, que afecte al proceso o producto de desarrollo de software y resumir en media página sus principales características.
<https://www.aenor.com/Buscador>
1. Poned el nombre de alguna herramienta de software que se pueda usar para gestión de calidad, justificando su utilidad.

Entrega individual