
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:** requisitos claros. Proceso planificado (realista) y basado en estándares. Uso y generación de documentación. Evaluación de objetivos y cumplimiento de compromisos. Supervisión y corrección.

3. Mejora de calidad

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

Basado en **medidas cuantitativas** (no demasiadas, ralentización). Se puede medir la productividad, el esfuerzo, los defectos detectados, etc.

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

La mejora implica un cambio. Motivos de la resistencia al cambio:

No estar de acuerdo, incertidumbre respecto a roles y tareas, pérdida de estatus, riesgo mayor que continuar como se está.

En muchos casos se requiere una **estimación** de los costes que supondrá realizar el cambio (modelos de estimación).

Estrategias de Mejora

- Acuerdo de mejoras con autoridad y líderes para tener guías, tener claros los objetivos del cambio
- Alinear mejoras con los requisitos.
- Trabajar de forma incremental, priorizando requisitos en las iteraciones. Para cada problema a resolver: qué costes supone el problema, qué cambiar, cómo hacer el cambio, estimar costes y esfuerzo que supondrá el cambio, evaluación real tras el cambio, análisis y comparativa.
- Realización de informes de resultados y avances, informar a todos los implicados, aprender de cada iteración.
- Justificar decisiones, escalar a un rol superior los problemas que excedan a nuestras responsabilidades.
- Aprovechar lo positivo para que el equipo apoye el cambio: aprendizaje, organización, reconocimiento del trabajo, utilidad personal, gusto profesional.

Plan de mejora de proceso: ejemplo

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

Decisión 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.

número de partes de quejas, número de líneas de código, tiempo en que se hacen las quejas, esfuerzo (pm) y **estimaciones** de costes y esfuerzo,...

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.



Estándares de Facto: ITIL, COBIT

ITIL: guía de buenas prácticas y recomendaciones para la administración de servicios de Tecnologías de la Información (TI).

COBIT: framework con buenas prácticas que asiste a la empresa para lograr sus objetivos de gobernanza y gestión de IT.

ITIL (IT Infrastructure Library)

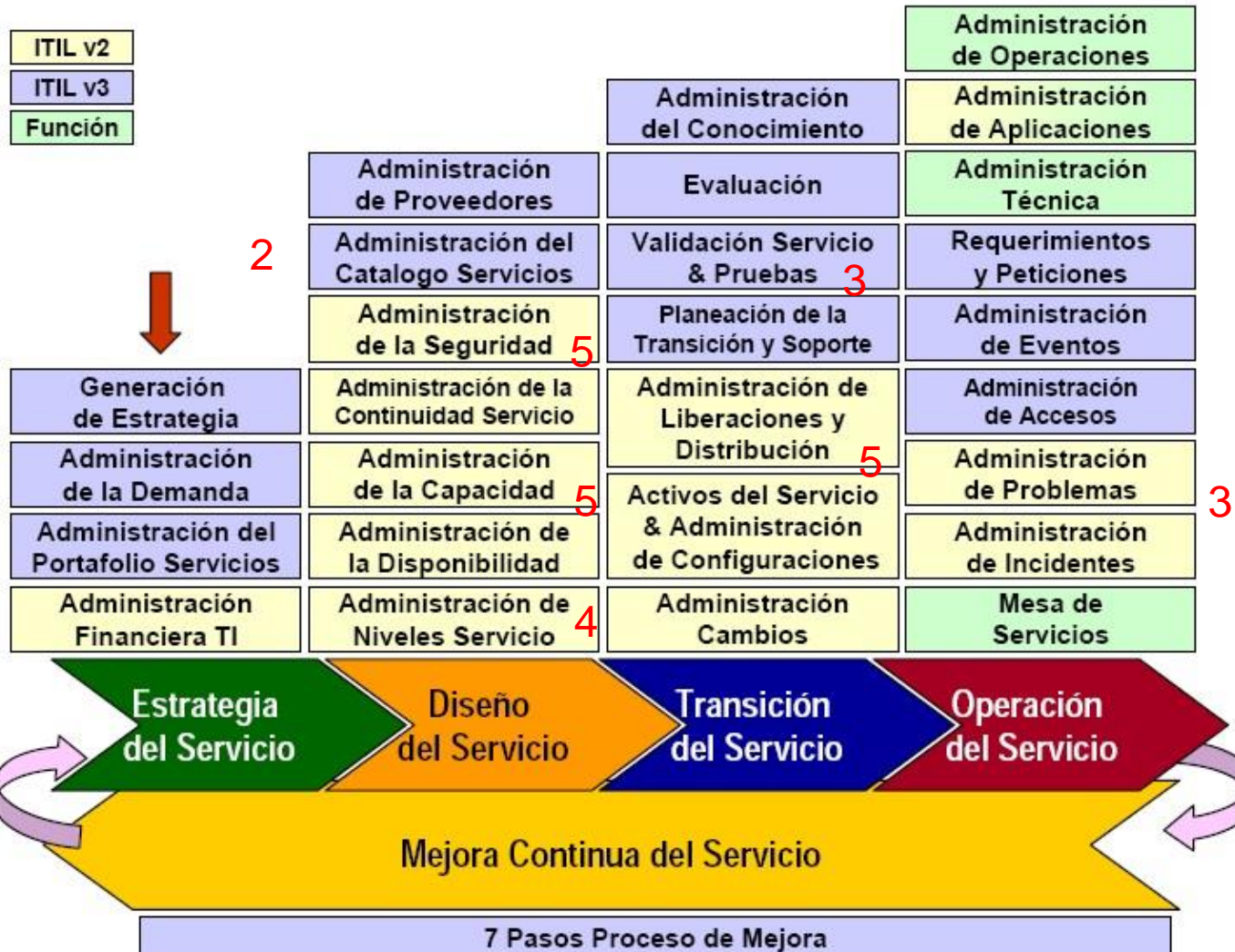
Marco de referencia que describe un conjunto de mejores prácticas y recomendaciones para la administración de servicios de Tecnologías de la Información (TI), con un enfoque de administración de procesos.

Lograr la integración eficiente de gente, procesos y tecnología para una mejor administración de los servicios de TI (no equipos ni tecnología directamente sino los servicios ofrecidos a los clientes), optimizando el uso de los recursos y mejorando constantemente la calidad de los servicios.

Una persona puede obtener una certificación en ITIL que acredite sus conocimientos sobre este marco de referencia.

ITIL (IT Infrastructure Library)

Fuente: Plutorak & Associates, Ltd



Los procesos sirven para planificar, entregar y dar soporte a los servicios.

Orden sugerido para implantación:

1. Planificar modo
2. Catálogo de servicios IT
3. Estabilizar
4. Mejorar
5. Mejora continúa

ITIL (IT Infrastructure Library)

Una organización puede intentar seguir las guías de ITIL y a la vez conseguir que a nivel regulatorio se **alinee a otros estándares y mejores prácticas** como ISO-20000, CobiT, ISO-27001, etc. No hay incompatibilidad entre ellos, todos pueden coexistir y complementarse.

La implementación de ITIL requiere:

- Casi siempre un **cambio a nivel organizacional** (las personas, sus funciones y sus relaciones).
- **Implicación de la alta gerencia** dando apoyo y financiando los gastos asociados (incluidos nuevos recursos y capacitación específica de personal).
- **Participación de todo el personal en los cambios** que se vayan decidiendo para alinear sus procesos a los de ITIL.
- **Paciencia, tiempo, constancia y método.**

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 como CMMI, ISO/IEC 20000, ISO 9000,...



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 y cada cierto tiempo se tiene que renovar.

CMMI (Capability Maturity Model Integration)

Modelo para planificar y evaluar la realización de un proyecto con buenas prácticas.

Aplicable a grandes organizaciones (difícil de aplicar a medianas o pequeñas).

Deja en manos de los desarrolladores la implementación de muchas de las tareas.

- Evalúa 5 **niveles de madurez** de los procesos, la empresa debe aspirar al último:
 - Inicial
 - Repetible
 - Definido
 - Administrado
 - Optimización

CMMI

- En cada nivel, puede haber 5 estados en los procesos:
 - (1) mal, descontrolado o impredecible
 - (2) hay acuerdos sobre los procesos
 - (3) hay adecuación tecnológica
 - (4) hay adecuación organizativa
 - (5) hay mejora continua en base a indicadores

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.

<http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0048609#.VHCJB8kbi1s>

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.

ISO/IEC 20000

Implica a los procesos de:

- **Provisión de servicio:** requisitos o alcance, presupuestos, seguridad, disponibilidad, informes
- **Control:** gestión de configuración (versiones), cambios
- **Entrega**
- **Resolución:** incidentes y problemas
- **Relaciones** del negocio con clientes y suministradores

Para cada proceso se establecen unos objetivos, los responsables y las buenas prácticas que deben realizar para que se realice con calidad y mejore. Se dice qué hacer pero no se concreta cómo.

ISO/IEC 20000

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

http://www.iso.org/iso/catalogue_detail.htm?csnumber=51988

- 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

ISO/IEC 9126

Norma para especificar la **calidad de un producto**.

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

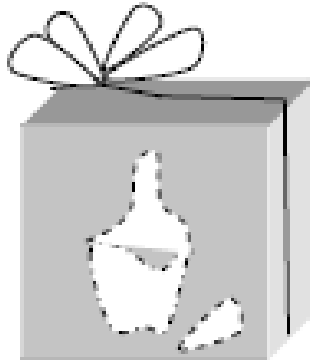
Calidad interna y externa: Orientada a evaluar el cumplimiento las siguientes características en los productos software (RNF):

- **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.
- Controlar la documentación del software y de los cambios realizados.
- 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

Son **reuniones de trabajo periódicas**, realizadas por los responsables de un proyecto, 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

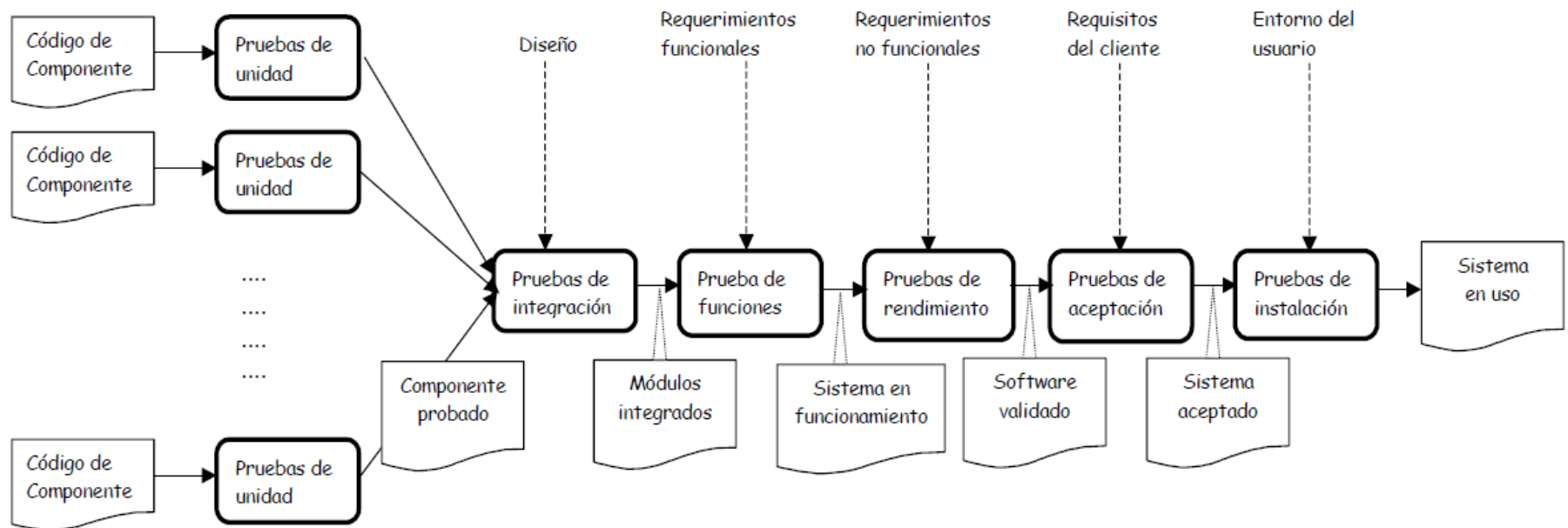
Pruebas

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



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. Resumir en unas líneas la finalidad de COBIT y exponer en una página un ejemplo de buena práctica considerada por este framework.
2. Buscar una oferta de trabajo relacionada con la calidad del software.

Entrega individual.