Calidad del software

- •Todas las metodologías y herramientas tienen un único fin: **producir software de gran calidad**
- •Definiciones de calidad del software
- "Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo **software** desarrollado profesionalmente" R. S. Pressman (1992).
- "El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas" ISO 8402 (UNE 66-001-92).
- Conclusiones
- Los requisitos del software son la base de las medidas de calidad.
 La falta de concordancia con los requisitos es una falta de calidad
 Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad
- Existen algunos requisitos implícitos o expectativas que a menudo no se mencionan, o se mencionan de forma incompleta (por ejemplo el deseo de un buen mantenimiento) que también pueden implicar la falta de calidad.

<u>Aseguramiento de calidad del software</u> (<u>Software Quality Assurance - SQA</u>)

El **aseguramiento** de **calidad** del **software** *es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad.*

El **aseguramiento** de **calidad** del **software** se diseña para cada aplicación antes de comenzar a desarrollarla y no después.

Algunos autores prefieren decir **garantía** de **calidad** en vez de **aseguramiento**.

- Garantía, puede confundir con garantía de productos
- Aseguramiento pretende dar confianza en que el producto tiene calidad

El aseguramiento de calidad del software está presente en:

- Métodos v herramientas de análisis, diseño, programación v prueba
- Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software
- Estrategias de prueba multiescala
- Control de la documentación del **software** y de los cambios realizados
- Procedimientos para ajustarse a los estándares (y dejar claro cuando

se está fuera de ellos)

- Mecanismos de medida (métricas)
- Registro de auditorias y realización de informes

Actividades para el aseguramiento- de calidad del software

- Métricas de **software** para el control del proyecto
- Verificación y validación del **software** a lo largo del ciclo de vida
 Incluye las pruebas y los procesos de revisión e inspección
- La gestión de la configuración del software

<u>Gestión de la calidad del software</u> (Software Quality Management - SCM)

Gestión de la calidad (ISO 9000)

– Conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad.

Política de calidad (ISO 9000)

 Directrices y objetivos generales de una organización, relativos a la calidad, tal como se expresan formalmente por la alta dirección

La gestión de la **calidad** se aplica normalmente a nivel de empresa

También puede haber una gestión de **calidad** dentro de la gestión de cada proyecto

Control de la calidad del software (Software Quality Control - SQC)

Son las técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la **calidad**, centradas en dos objetivos fundamentales:

- mantener bajo control un proceso
- eliminar las causas de los defectos en las diferentes fases del ciclo de vida

En general son las actividades para evaluar la **calidad** de los productos desarrollados

Sistema de calidad

 Estructura organizativa, procedimientos, procesos y recursos necesarios para implantar la gestión de calidad

El sistema de **calidad** se debe adecuar a los objetivos de **calidad** de la empresa

La dirección de la empresa es la responsable de fijar la política de **calidad** y las decisiones relativas a iniciar, desarrollar, implantar y actualizar el sistema de **calidad**.

Un sistema de calidad consta de varias partes:

- Documentación:
- Manual de **calidad**. Es el documento principal para establecer e implantar un sistema de **calidad**. Puede haber manuales a nivel de empresa, departamento, producto, específicos (compras, proyectos,...)
- Parte física: locales, herramientas ordenadores, etc.
- Aspectos humanos:
- Formación de personal
- Creación y coordinación de equipos de trabajo

Normativas:

- ISO 9000: Gestión y **aseguramiento** de **calidad** (conceptos y directrices generales)
- Recomendaciones externas para aseguramiento de la calidad (ISO 9001, ISO 9002, ISO 9003)
- Recomendaciones internas para aseguramiento de la calidad (ISO 9004)
- MALCOM BALDRIGE NATIONAL QUALITY AWARD
- Software Engineering Institute (SEI) Capability Maturity Model (CMM) for software

Certificación de la calidad (Quality certification - QC)

Un sistema de certificación de **calidad** permite una valoración independiente que debe demostrar que la organización es capaz de desarrollar productos y servicios de **calidad**

Los pilares básicos de la certificación de calidad son tres :

- Una metodología adecuada
- Un medio de valoración de la metodología
- La metodología utilizada y el medio de valoración de la metodología deben estar reconocidos ampliamente por la industria

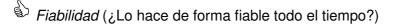
Factores que determinan la calidad del software

Se clasifican en tres grupos:

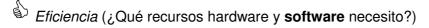
Operaciones del producto: características operativas



El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente



El grado que se puede esperar de una aplicación lleve a cabo las operaciones especificadas y con la precisión requerida



La cantidad de recursos hardware y **software** que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados

Integridad (¿Puedo controlar su uso?)

El grado con que puede controlarse el acceso al **software** o a los datos a personal no autorizado

Facilidad de uso (¿Es fácil y cómodo de manejar?)

El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados

Revisión del producto: capacidad para soportar cambios

Facilidad de mantenimiento (¿Puedo localizar los fallos?)

El esfuerzo requerido para localizar y reparar errores

Flexibilidad (¿Puedo añadir nuevas opciones?)

El esfuerzo requerido para modificar una aplicación en funcionamiento

Facilidad de prueba (¿Puedo probar todas las opciones?)

El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos

Transición del producto: adaptabilidad a nuevos entornos

Portabilidad (¿Podré usarlo en otra máquina?)

El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo

Reusabilidad (¿Podré utilizar alguna parte del **software** en otra aplicación?)

Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones

Interoperabilidad (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?

El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos

Métricas de la calidad del sofware

Es difícil, y en algunos casos imposible, desarrollar medidas directas de los factores de **calidad** del **software**

Cada factor de **calidad** Fc se puede obtener como combinación de una o varias métricas:

 $F_c = c_1 * m_1 + c_2 * m_2 + ... + c_n * m_n$

- C_i factor de ponderación de la métrica i, que dependerá de cada aplicación específica
- mi métrica i

Habitualmente se puntúan de 0 a 10 en las métricas y en los factores de calidad

Métricas para determinar los factores de calidad:

- Facilidad de auditoria
- Exactitud
- Normalización de las comunicaciones
- Completitud
- Concisión
- Consistencia
- Estandarización de los datos
- Tolerancia de errores
- Eficiencia de la ejecución
- Facilidad de expansión
- Generalidad
- Independencia del hardware
- Instrumentación
- Modularidad
- Facilidad de operación
- Seguridad
- Autodocumentación
- Simplicidad
- Independencia del sistema
- Facilidad de trazabilidad
- Formación