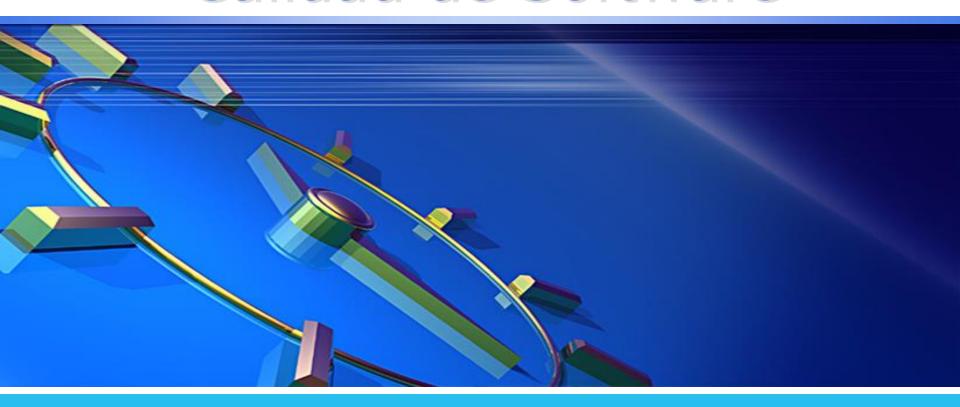


Tema 1. Fundamentos de Calidad de Software



Contenido



Introducción

"I do not worry whether something is cheap or expensive. I only worry if it is good. If it is **good enough**, the public will pay you back for it"

Walt Disney

No me preocupa si algo es barato o caro. Solo me preocupa si es bueno. Si es lo suficientemente bueno, el público te lo pagará.

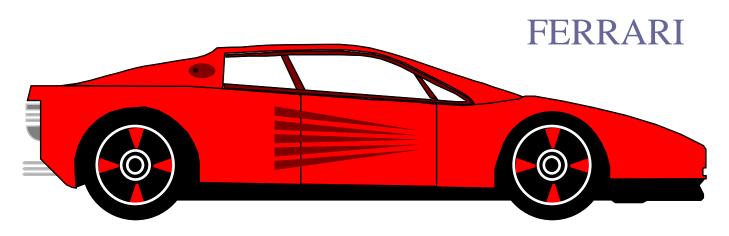
Que tiene mas calidad

Los dos tienen la misma calidad siempre y cuando cumplan con sus requerimientos





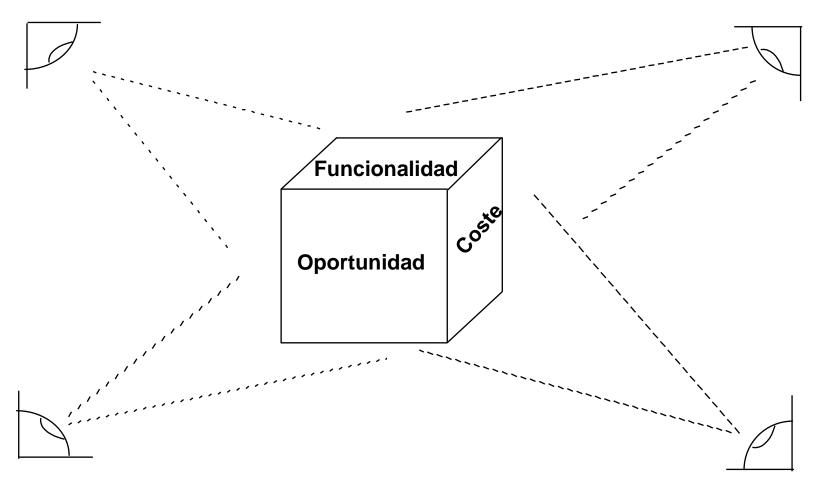
Para ello debemos probar sus especificaciones



La calidad es relativa a las personas, a su edad, a las circunstancias de trabajo, el tiempo...

FIAT

- •Un caramelo para un niño.
- •Un mapa gastronómico mundial.
- •El tiempo varia las percepciones.



VISTAS DE LA CALIDAD

Garvin (1984)

- TRASCENDENTAL (calidad = excelencia innata)
- * BASADA EN USUARIO (adecuación al propósito)
- BASADA EN FABRICANTE (conformidad con requisitos)
- * BASADA EN PRODUCTO (económica)
- * BASADA EN VALOR (precio asequible)

Concepto de calidad: Definiciones

- ❖ Propiedad o conjunto de propiedades inherentes a una cosa, que permiten apreciarla cono igual, mejor o peor que las restantes de su especie (DRAE).
- ❖ Totalidad de las características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implícitas (Norma UNE 66-001-92 traducción de ISO 8402).

- Adecuación (del producto) al uso (Juran)
- Conformidad con requisitos y confiabilidad en el funcionamiento (Deming)
- Cero defectos (Crosby)
- Pérdida económica que un producto supone para la sociedad desde el momento de su expedición (Taguchi)
- Grado en el que un conjunto de características inherentes cumple con los requisitos (ISO 9000:2000)

- Totalidad de las características y aspectos de un producto o servicio en los que se basa su aptitud para satisfacer una necesidad dada (EOQ)
- El grado de satisfacción que produce al cliente
- Un buen producto no es el que cumple con una determinada especificación, sino el que es bien recibido por el cliente (Drucker)

CONCEPTO DE CALIDAD

Gillies (1992)

- No es absoluto
- Está sujeto a restricciones
- Trata de compromisos aceptables
- Es multidimensional
- Los criterios de calidad no son independientes

Concepto de calidad

❖ Según la UNE 66-001-92 [AENOR, 1992], se define la calidad como: "Totalidad de características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades expresadas o implicitas"

La consecución de la calidad puede tener tres orígenes:

- Calidad Realizada: La que es capaz de obtener la persona que realiza el trabajo.
- Calidad Programada: La calidad que se ha pretendido obtener.
- Calidad Necesaria: La calidad que el cliente exige con mayor o menor grado de concreción.



Calidad del Software

- Hay que tener en cuenta a la hora de abordar la calidad en el software un conjunto de características del mismo que lo hace un producto peculiar:
 - Se desarrolla, no se fabrica en el sentido clásico del mismo.
 - Se trata de un producto lógico, sin existencia física.
 - No se degrada con el uso.
 - Por la complejidad del SW y la ausencia de controles adecuados, se suele entregar el SW conscientemente con defectos (incluso públicamente declarados).
 - Un gran porcentaje de la producción se hace aún a medida en vez de emplear componentes existentes y ensamblar.
 - Es muy flexible. Se puede cambiar con facilidad e incluso reutilizar fragmentos.

Definición de calidad del software

Definición oficial (IEEE Std. 610-1990) Es el grado con el que un sistema, componente o proceso cumple:



- Los requisitos especificados.
- Las necesidades o expectativas del cliente o usuario.

Concordancia del software producido 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.

- Los requisitos establecidos explícitamente se reflejan en el documento de especificación de requisitos del sistema:
 - •Funcionales: funciones a realizar por el software.
 - •No funcionales (o extendidos): requisitos de seguridad, de rendimiento, etc...
- ❖ Los requisitos implícitos no aparecen en el documento de especificación de requisitos del sistema. Si se cumplen los explícitos y no los implícitos, la calidad del software queda en entredicho.
- El uso de estándares y las normas de desarrollo permiten que se consiga una calidad técnica.

Tópicos relacionados con la Calidad (i)

Gestión de la calidad del Software

- Aspectos de la función general de la gestión que determina y aplica la política de calidad (objetivos y directrices generales de calidad de una empresa). Incluye:
 - · Planificación estratégica.
 - Asignación de recursos.
- Puede haber una gestión de la calidad dentro de cada proyecto.

* Aseguramiento de la calidad del software

- Conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (SW) satisfará los requisitos dados de calidad.
- Conjunto de actividades para evaluar el proceso mediante el cual se desarrolla el producto

Tópicos relacionados con la Calidad (ii)

Control de calidad del software

- 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 y eliminar las causas de defectos en las diferentes fases del ciclo de vida.
- Proceso de verificar el propio trabajo o el de un compañero.

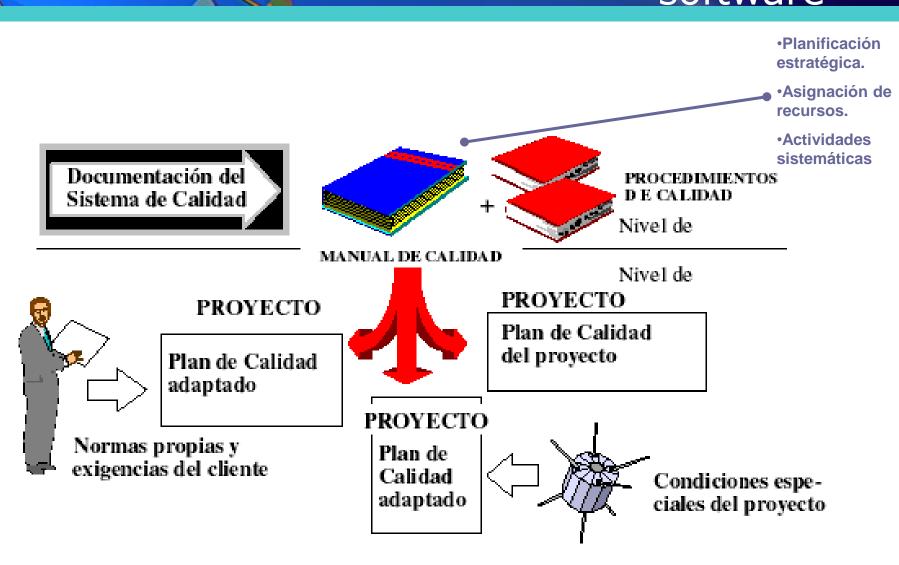
Verificación o validación del SW: Actividad ligada al control de la calidad en el ámbito del software

- Verificación: Comprobar si los productos construidos en una fase del ciclo de vida satisfacen los requisitos.
- Validación: Comprobar si el software construido satisface los requisitos de usuario.

Niveles de acción en la ingeniería del software

- El trabajo para la mejora de la calidad tiene distintos ámbitos de actuación:
 - Nivel individual
 - Nivel de empresa/organización
 - Nivel de proyecto
- La gestión de la calidad a nivel de empresa u organización consiste en la creación de una estructura organizativa apropiada para fomentar el trabajo por la calidad de todas las personas y departamentos de la empresa. Se suele recurrir al concepto de sistema de calidad
- ❖ El desarrollo del software se suele organizar en proyectos. En cada proyecto de desarrollo se deben aplicar las directrices de calidad fijadas a nivel de la organización. Para ello es imprescindible la adaptación de las mismas a las condiciones de cada proyecto. Las directrices contenidas en el sistema de calidad deben adecuarse a cada uno de los proyectos.

Niveles de acción en la ingeniería del software



*Para la implementación de una infraestructura de calidad es necesario el apoyo de un sistema de calidad que se adecue a los objetivos de calidad de la empresa, porque es un punto vital:

Estructura de organización, de responsabilidades, de actividades, de recursos y de procedimientos que se establecen para llevar a cabo la gestión de la calidad. ISO-9000

❖Este sistema debe adecuar los objetivos de la calidad a de la empresa.

*La dirección es la responsable de:

- Fijar la política de la calidad
 - "un 95% de los trenes llegan con de 5 min. de retraso"
 - "el cliente sismpre tiene la razón"
- Las decisiones relativas al inicio, desarrollo, implantación y actualización del sistema de calidad.

- ❖ Se debe fijar la estructura organizativa al sistema de gestión de calidad (líneas jerarquicas y de comunicación.
- Para se útil, un sistema de calidad debe:
 - Ser eficaz, comprendido por todos
 - Ofrecer confianza en satisfacer las necesidades de los clientes.
 - Poner énfasis en prevenir en lugar de detectar.

Un sistema de calidad consta de dos partes:

- <u>Documentación</u>: en la que se describe el sistema, procedimientos, etc. ajustándose a una norma:
 - Manual de calidad: Descripción del sistema que sirve de referencia permanente en la aplicación del sistema.
 - *Procedimientos de calidad*: Instrucciones específicas para ciertas actividades o procesos.
 - Registros de datos sobre calidad: Almacenamiento de información sobre actividades relacionadas con la calidad.
- Parte practica, que tiene dos vertientes:
 - Aspectos físicos (locales, herramientas, ordenadores,...)
 - Aspectos humanos: formación del personal a todos los niveles y creación y coordinación de equipos de trabajo.

Manual de calidad

- Los elementos, requisitos y los medios que adopte la empresa para su sistema de calidad se deben establecer por escrito, ordenadamente, en forma de políticas y procedimientos.
- Debe describir el sistema de gestión de calidad para servir como referencia al implantar el sistema. En grandes empresas:
 - Puede realizarse para la totalidad de la empresa
 - Puede haber manuales a nivel de departamento, producto, etc.
 - Puede haber manuales específicos (compras, desarrollos/proyectos, etc.)

Procedimientos

- Para que el manual sea más manejable, puede completarse con procedimientos o instrucciones específicas para ciertas actividades o procesos.
- Cada empresa puede tener sus propios procedimientos, que suelen fundamentarse en:
 - La buena práctica y el saber hacer.
 - Los códigos, las normas y las especificaciones a los que deben ajustarse

* Registros de datos sobre calidad

- Pretenden almacenar datos sobre las actividades relacionadas con la calidad o sobre la evaluación de los productos:
 - Datos de pruebas
 - Datos sobre revisiones
 - Inspecciones
 - Datos de costes, actividades
 - etc

Calidad a nivel de proyecto

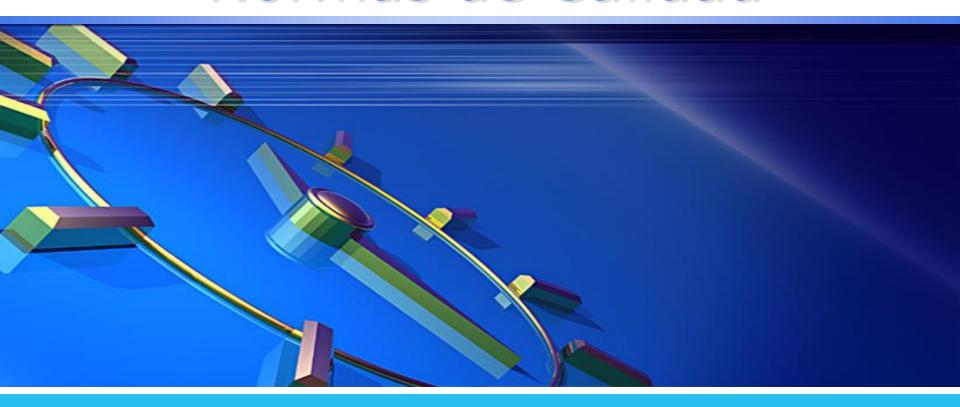
- ❖ Para adaptar las directrices marcadas por los sistemas de calidad a cada proyecto particular, hay que generar un plan específico de calidad: Plan de aseguramiento de la calidad. El plan de aseguramiento debe contener:
 - Objetivos de calidad del proyecto y enfoque para su consecución
 - Documentación referenciada en el plan
 - Gestión del aseguramiento de la calidad
 - Documentación de desarrollo y de control o gestión
 - Estándares, normas y prácticas que hay que cumplir
 - Actividades de revisión y auditorias

Calidad a nivel de proyecto

- Gestión de la configuración del software
- Informes de problemas
- Herramientas, técnicas y métodos de apoyo
- Control del código, de los equipos y de los suministradores
- Recogida, mantenimiento y almacenamiento de datos sobre la documentación de las actividades de aseguramiento de la calidad realizadas



Tema 2. Modelos y Normas de Calidad



Contenido



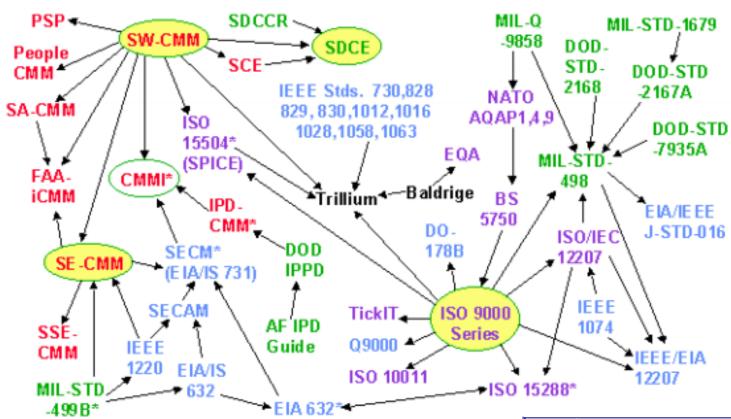
Estandarización Internacional

Concepto de Calidad

Conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer una necesidades explícitas o implícitas (ISO 8402)

Actividades de estandarización internacional

Existen varias organizaciones de estandarización internacional, algunas son regionales mientras que otras son globales. Las últimas están relacionadas con la ONU o son independientes, como por ejemplo la International Telecomunication Union (ITU). El ISO y el IEC son de particular importancia para SPICE.



* Not yet released

Copyright @ 1995, Software Productally Conscribut NFP, hc. All rights reserved.

Color	Significado
Rojo	Un Modelo de Capacidad de Madurez
Verde	Un documento de gobierno o milicia de Estados Unidos
Morado	Un estándar Internacional
Azul	Un documento de una asociación industrial, profesional o comercial (en su mayoría de Estados Unidos)
Vegro	Otro

Actividades de estandarización

La International Electrotechnical Commission o Comisión Electrónica Internacional (IEC) fué fundada en el año 1906 para definir estándares en eléctrica y electrónica, mientras que la International Organization for Standarization (ISO) fué creada en 1947 para abarcar otros temas.

Actividades de estandarización

La International Electrotechnical Commission o

Comisión Electrónica Internacional (IEC) fué
fundada en el año 1906 para definir estándares
en eléctrica y electrónica

Ambas tienen por objetivo facilitar intercambio de bienes y servicios a nivel internacional que abarcan diversas áreas de IT.

Actividades de estandarización

En 1987, ISO e IEC decidieron formar el Joint
Technical Committe o Comité Técnico
Conjunto

(JTC), cuyo objetivo es elaborar estándares para la Tecnología de Información o Information Technology (IT).

Principales normas ISO

- ISO 216 Medidas de papel: p.e. ISO A4
- ISO 639 Nombres de lenguas
- **ISO 690:1987** regula las citas bibliográficas (corresponde a la norma UNE 50104:1994)
- **ISO 690-2:1997** regula las citas bibliográficas de documentos electrónicos
- ISO 732 Formato de carrete de 120
- ISO 838 Estandar para perforadoras de papel
- ISO 1007 Formato de carrete de 135
- ISO/IEC 1539-1 Lenguaje de programación Fortran
- ISO 3029 Formato carrete de 126

- ISO 3166 códigos de países
- ISO 4217 códigos de divisas
- ISO 7811 Técnica de grabación en tarjetas de identificación
- ❖ ISO 8601 Representación del tiempo y la fecha. Adoptado en Internet mediante el Date and Time Formats de W3C que utiliza UTC.
- ISO 8859 codificaciones de caracteres que incluye ASCII como un subconjunto (Uno de ellos es el ISO 8859-1 que permite codificar las lenguas originales de Europa occidental, como el español)
- * ISO/IEC 8652:1995 Lenguaje de programación Ada
- ISO 9000 Sistemas de Gestión de la Calidad -Fundamentos y vocabulario
- * ISO 9001 Sistemas de Gestión de la Calidad Requisitos
- ISO 9004 Sistemas de Gestión de la Calidad Directrices para la mejora del desempeño

- ISO 9660 Sistema de archivos de CD-ROM
- * ISO 9899 Lenguaje de programación C
- ISO 10279 Lenguaje de programación BASIC
- * ISO 10646 Universal Character Set
- *** ISO/IEC 11172** MPEG-1
- ❖ ISO/IEC_12207 Tecnología de la información / Ciclo de vida del software
- * ISO 13450 Formato de carrete de 110
- *** ISO/IEC 13818** MPEG-2

- ISO 14000 Estándares de Gestión Medioambiental en entornos de producción
- ISO/IEC 14496 MPEG-4
- **ISO/IEC 15444** JPEG 2000
- ISO 15693 Estándar para "tarjetas de vecindad"
- ISO/IEC 17799 Seguridad de la información
- ISO 26300 OpenDocument
- ISO/IEC 17025 Requisitos generales relativos a la competencia de los laboratorios de ensayo y calibración
- ISO/IEC 27001 Sistema de Gestión de Seguridad de la Información

Con el objetivo de estandarizar los sistemas de calidad de las diferentes empresas y sectores, se publican las normas ISO 9000, que son un conjunto de normas editadas y revisadas periódicamente por la Organización Internacional de Normalización (ISO) sobre la garantía de calidad de los procesos.

Así, se consolida a nivel internacional la normativa de la gestión y control de calidad.

- Publicada el año 1987. Adoptada por más de 90 paises.
- Directrices para la gestión del sistema de calidad y modelos de garantía de calidad para la empresa.
- Las directrices son genéricas y aplicables a cualquier sector.
- Es un marco de trabajo para la mejora continua.

ISO 9000 Modelo de Calidad Total

ISO	TÍTULO
8402	Vocabulario - Terminología.
9000	Normas para la gestión y garantía de la calidad. Directrices de selección y uso (ISO 9000-1 1.994). Directrices generales para aplicar las normes 9001, 9002, 9003 (ISO 9000-2 1.993). Guía para aplicar las normes 9001 a empresas de software (ISO 9000-3 1.996). Guía para la gestión de un programa de seguridad (ISO 9000-4).
9001	Modelo para la garantía de la calidad en diseño / desarrollo, producción, instalación y servicio.(1.994)
9002	Modelo para garantizar la calidad en producción y servicios. (1.994)
9003	Modelo para garantizar la calidad en inspección final y pruebas. (1.993)
9004	Elementos y gestión del sistema de calidad. Reglas generales. Directrices para los serveis (ISO 9004-2). Directrices para materiales procesados (ISO 9004-3). Directrices para la mejora de la calidad (ISO 9004-4).

Objetivos de ISO 9000:

- Proporcionar una guía para la gestión de la calidad: diseño e implantación de sistemas de calidad.
 - (ISO 9000 no normaliza el sistema de gestión de calidad, ya que esto depende del tipo de sector, tamaño de la empresa, organización interna, etc, sino que normaliza las verificaciones que se han de realizar sobre el sistema de calidad)
- Describir los requerimientos generales para garantizar la calidad (demostrar la idoneidad del sistema de calidad).

ISO 9000 (Recomendaciones)

- Comenzar con ISO 9004-1(1994). Gestión de calidad y elementos del sistema de calidad, para diseñar y implementar el sistema de gestión de calidad.
 - ISO 9004 es la directriz para el establecimiento o ampliación de un sistema de calidad. Esta norma amplia partes de ISO 9000 que no se pueden verificar o que una parte contratante no desea dar a conocer, como por ejemplo los gastos asociados a la gestión de calidad.
- Una vez implantado el sistema de calidad, utilizar los modelos de garantía de calidad ISO 9001-2000 para demostrar su idoneidad

\$9,9000 Aspectos positivos

- Es un factor competitivo para las empresas
- Proporciona confianza a los clientes
- Ahorra tiempo y dinero, evitando recertificar la calidad según los estándares locales o particulares de una empresa.
- Se ha adaptado a más de 90 países e implantado a todo tipo de organizaciones industriales y de servicios, tanto sector privado como público
- Proporciona una cierta garantía de que las cosas se hacen tal como se han dicho que se han de hacer

9000 Aspectos negativos

- Es costoso
- Muchas veces se hace por obligación.
- Es cuestión de tiempo que deje de ser un factor competitivo
- Hay diferencias de interpretación de las cláusulas del estándar
- No es indicativa de la calidad de los productos, procesos o servicio.
- Hay mucha publicidad engañosa.

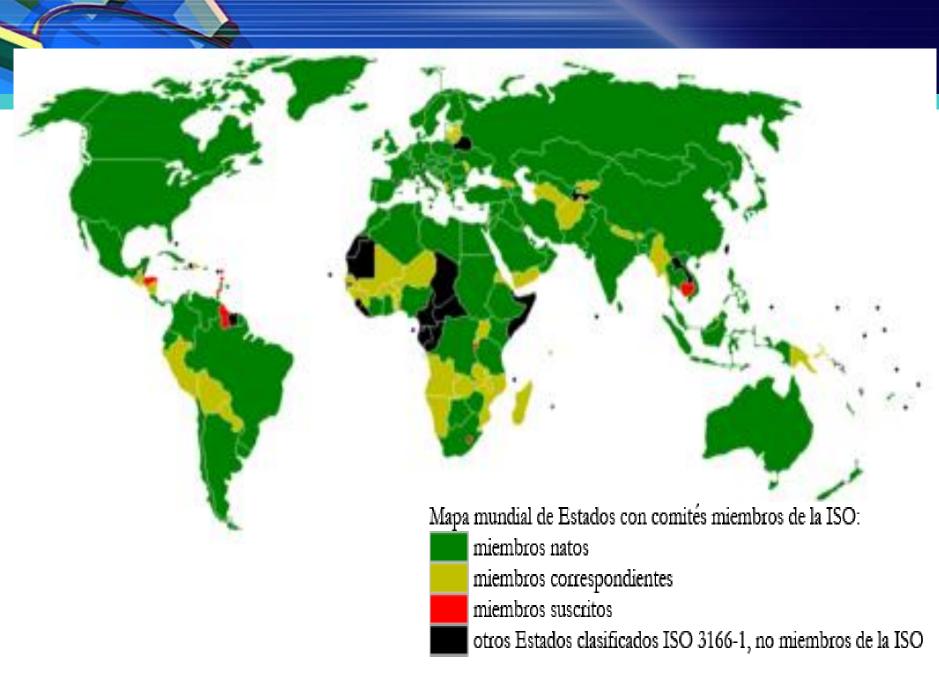
- En 1997 había 4605 empresas certificadas.
- Sólo 66 referentes a actividades informáticas
- En Baleares no hay ninguna empresa certificada que se dedique a actividades informáticas
- 56 de las 66 certificaciones de actividades informáticas fueron certificadas por AENOR (Asociación Española de NORmalización).

ISO 9000: Calidad de Software

- ❖ ISO 9001:2000. Modelo para conseguir la calidad total en el diseño, desarrollo, producción, instalación y servicio post-venta.
- ❖ ISO 9000-3:1991. Guía para la aplicación de la norma ISO 9001 al desarrollo, suministro y mantenimiento de software.
 - En todo caso, nos certificaríamos según ISO 9000-3.
 - No añade ni cambia los requerimientos de la ISO 90001. Los amplia y aclara.
- Otras normas aplicables
 - ISO 9004-1:1994. Gestión de la calidad y elementos del sistema de calidad (Guía per establecer el QA).

ISO 9000: Calidad de Software

- ISO 8402:1994. Gestión de la calidad y garantía de la calidad. Vocabulario.
- ISO 12207:1995. Procesos del ciclo de vida del software.
- ISO/IEC 9126:1991. Características de la calidad de un producto software.
- ISO/IEC 12119:1995. Productos software: evaluación y test.
- ISO/IEC 14102:1995. Guía para la evaluación y selección de herramientas CASE.



Estructura de la organización

La Organización ISO está compuesta por tres tipos de miembros:

Miembros natos

Uno por país, recayendo la representación en el organismo nacional más representativo.

Estructura de la organización

Miembros correspondientes

De los organismos de países en vías de desarrollo y que todavía no poseen un comité nacional de normalización. No toman parte activa en el proceso de normalización pero están puntualmente informados acerca de los trabajos que les interesen.

Estructura de la organización

Miembros suscritos

Países con reducidas economías a los que se les exige el pago de tasas menores que a los correspondientes.

Normas de Calidad del Proceso

ISO / IEC TR 15504



SPICE) ISO/IEC -TR 15504

Software Process Improvement Capability dEtermination

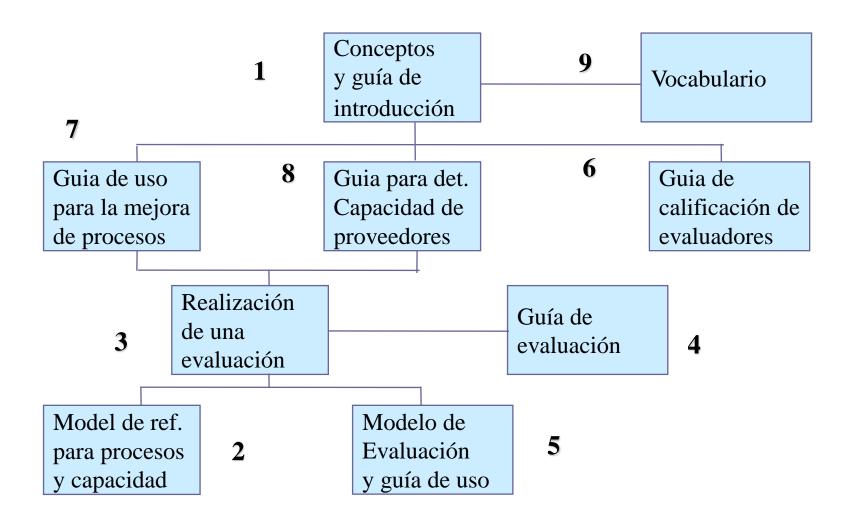
(Modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software).

- Evaluación y mejora de procesos software.
- Inicio del proyecto 1993
- Se halla en fase de Informe Técnico
- Es aplicable a cualquier organización o empresa que quiera mejorar la capacidad de cualquiera de sus procesos de software.
- Se puede utilizar como herramienta de evaluación del estado de los procesos de software de la empresa.
- Es independiente de la organización, modelo del ciclo de vida, metodología y tecnología.

SPICE

- Marco para métodos de evaluación, no un método o modelo en sí
- Abarca:
 - Evaluación de procesos
 - Mejora de procesos
 - Determinación de capacidad
- Alineado con el ISO/IEC 12207
- Intenta proporcionar un marco en el que armonizar los enfoques existentes
- ❖ Se encuentra en la fase de Informe Técnico (TR) Tipo 2

Componentes de SPICE



2: Modelo de Referencia

- El modelo de referencia de SPICE describe los procesos que una organización puede realizar para comprar, suministrar, desarrollar, operar, mantener y soportar el software, así como los atributos que caracterizan la capacidad de estos procesos
- Proporciona una base para medir la capacidad de los procesos, en función de grado de consecución de sus atributos.
- El tiene dos dimensiones: Procesos y Capacidad

Dimensión Procesos

- Contiene los procesos que se han de evaluar. Se corresponden con los procesos del ciclo de vida del software, definidos al estándar ISO 12207:1995
- Se agrupan en categorías, en función del tipo de actividad al cual se aplican:
 - CUS: Cliente-Proveedor.
 - ENG: Ingeniería.
 - SUP: Soporte.
 - MAN: Gestión.
 - ORG: Organización.

Dimensión Procesos CUS

La categoría CUS está formada por procesos que afecta directamente al cliente, soportan el desarrollo y la transición del software al cliente y permiten la correcta operación y uso del producto y/o servicio software.

- CUS.1 Adquisición de productos software y/o servicios
- CUS.2 Establecimiento de contratos
- CUS.3 Identificar las necesidades del cliente
- CUS.4 Realizar auditorías y revisiones conjuntas.
- CUS.5 Entrega e instalación del software.
- CUS.6 Mantenimiento del software.
- CUS.7 Proporcionar servicio al cliente.
- CUS.8 Valorar la satisfacción del cliente.

Dimensión Procesos ENG

La categoria ENG està formada per procesos que directamente especifica, implementa o mantienen el producto software, su relación con el sistema y su documentación

- ENG.1 Análisis y diseño de requerimientos del sistema
- ENG.2 Análisis de requerimientos del software.
- ENG.3 Diseño del software.
- ENG.4 Construcción del software.
- ENG.5 Integración y pruebas del software.
- ENG.6 Integración y pruebas del sistema.
- ENG.7 Mantenimiento del software y del sistema.

Dimensión Procesos SUP

Está formada por procesos que dan soporte a cualquiera del resto de procesos (incluidos los SUP), en distintos puntos del ciclo de vida del software.

- SUP.1 Documentación
- SUP.2 Gestión de la configuración del software
- SUP.3 Garantía de calidad
- SUP.4 Resolución de problemas
- SUP.5 Realizar revisiones conjuntas

Qimensión Procesos MAN

Formada por procesos utilizados en la gestión de cualquier tipo de proyecto o proceso en el ciclo de vida del software.

- MAN.1 Gestionar el proceso.
- MAN.2 Gestionar el proyecto.
- MAN.3 Gestionar la calidad.
- MAN.4 Gestionar los riesgos.

Dimensión Procesos ORG

Formada por procesos que establecen los objetivos de negocio de la organización.

- ORG.1 Alineamiento de la organització.
- ORG.2 Establecimiento del proceso
- ORG.3 Evaluación del proceso
- ORG.4 Mejora del procés.
- ORG.5 Gestión de recursos humanos.
- ORG.6 Infraestructura.
- ORG.7 Reutilización

Dimensión capacidad

- Define una escala de medida para determinar la capacidad de cualquier proceso
- Consta de seis niveles de capacidad y nueve atributos de procesos
 - 0 Incompleto
 - 1 Realizado (Realización del proceso)
 - 2 Gestionado (Gestión de realización, Gestión de productos)
 - 3 Establecido (Definición de procesos, Recursos de procesos)
 - 4 Predecible (Medición de procesos, Control de procesos)
 - 5 En optimización (Cambio de procesos, Mejora continua)

Prácticas base

- Cada proceso tiene un conjunto de prácticas base asociadas
- Las prácticas base describen las actividades esenciales de un proceso específico
- La realización de las prácticas base indica el grado de alcance de la finalidad del proceso

Prácticas de gestión

- Cada atributo de proceso tiene un conjunto de prácticas de gestión asociadas
- Las prácticas de gestión son las que implementan o institucionalizan un proceso de una manera general
- La realización de las prácticas de gestión indica la consecución del atributo en esa instancia del proceso

Evaluación de atributos

- Los atributos de un proceso se evalúan con N (Not), P (Partially), L (Largely) y F (Fully), siendo:
 - N No alcanzado (0% a 15%)
 - Poca o ninguna evidencia de la consecución del atributo
 - P Parcialmente alcanzado (16% a 50%)
 - Evidencia de un enfoque sistemático y de la consecución del atributo. aunque algunos aspectos de la consecución pueden ser impredecibles
 - L Ampliamente alcanzado (51% a 85%)
 - Evidencia de un enfoque sistemático y de una consecución significativa del atributo. La realización del proceso puede variar en algunas áreas
 - F Totalmente alcanzado (86% a 100%)
 - Evidencia de un enfoque completo y sistemático y de la consecución plena del atributo

CMM

Capability Maturity Model



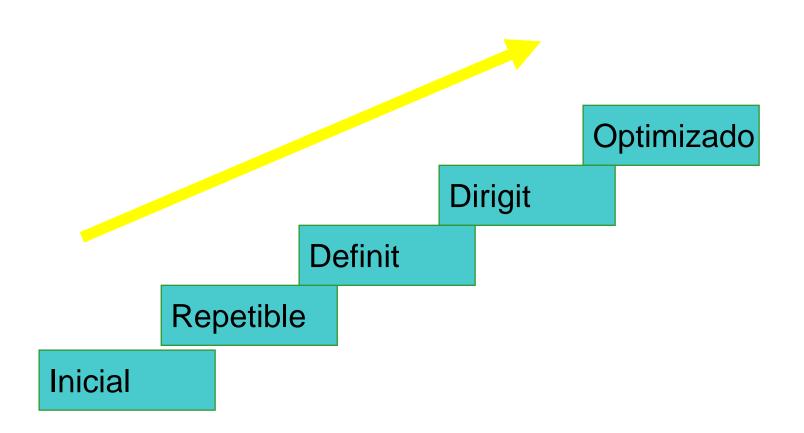
Software Engineering Institute Carnegy Mellon University

Mark C. Paulk

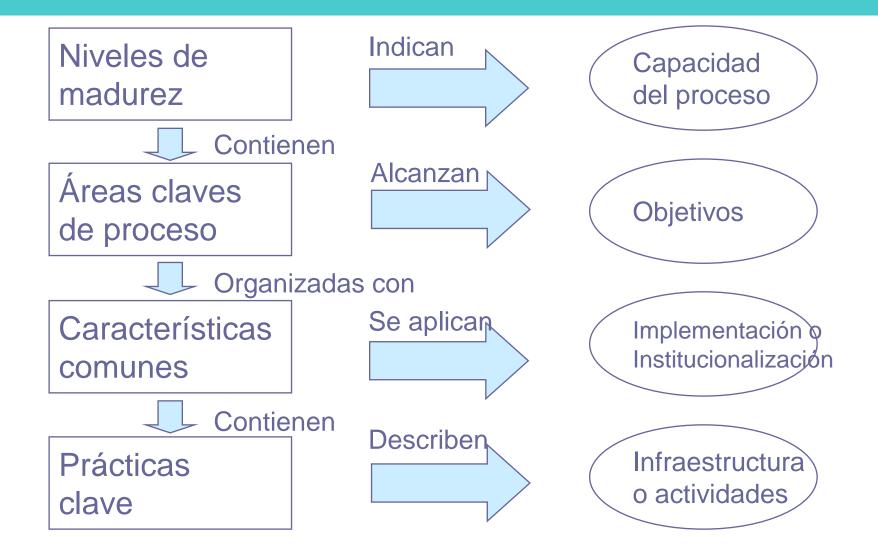
"CMM es una aplicación de sentido común de los conceptos de gestión de procesos y mejora de la calidad al desarrollo y mantenimiento del software"

- ❖ Estudia los procesos de desarrollo de software de una organización y produce una evaluación de la madurez de la organización según una escala de cinco niveles
- La madurez de un proceso es un indicador de la capacidad para construir un software de calidad.
- Es un modelo para la mejora de las organizaciones
- Obliga a una revisión constante.

CMM



CMM



- Es importante tener claro
 - Dónde nos encontramos
 - A dónde queremos llegar
 - Cómo llegaremos
 - Cómo sabremos si hemos llegado
- No se puede hacer todo de golpe
- Procesos piloto previos a un despliegue a gran escala.

La certificación, una exigencia?

- La Unión Europea edita el libro blanco sobre crecimiento, competitividad y puestos de trabajo, y reconoce la calidad como un elemento esencial de éxito de la empresa y constituye un factor estratégico en la política europea de competitividad.
- Las empresas precisan marcas y certificados que ayuden a vender sus productos en el mercado único en la era de la globalización.
- Se potencia la creación de infraestructuras de calidad: entidades de acreditación, organismos de normalización, entidades de inspección, etc.

La certificación, una exigència?

- Se impulsa la implantación de programas de calidad en las distintas administraciones públicas.
- Las grandes empresas exigen certificados de calidad a sus proveedores.
- Desde la administración se potencia mediante subvenciones, la implantación de programas de calidad.

Proceso habitual de certificación

- Motivación.
- Selección de la norma aplicable
- Subcontratación a empresa externa.
- Auditoría de certificación.
- Informe de acciones correctoras.
- Certificado.
- Imposición de seguimiento
- Incumplimiento!!!

Documentación

- Solicitud formal.
- Sistema de calidad
 - Manual de calidad
 - Manual de procedimientos.
 - Manual de especificaciones
 - Otros documentos
 - Expediente y certificación.

Otros aspectes

- Plazos y costes
 - Consultoría
 - Formación
 - Organismo certificador
- Mantenimiento de la certificación
 - Seguimiento anual.
 - Revisión de la certificación.

Normas de Calidad del Producto Software

- Introducción
- Modelo de calidad de producto Software: ISO 9126
- Evaluación del producto software: ISO 14598
- Ejemplos

Las evaluaciones deberían basarse en evidencias directas del producto, y no en evidencias circunstanciales del proceso"

Maibaum, T. y Wassyng, A. 2008. A Product-Focused Approach to Software Certification. Computer Volume: 41, Issue: 2: 91-93

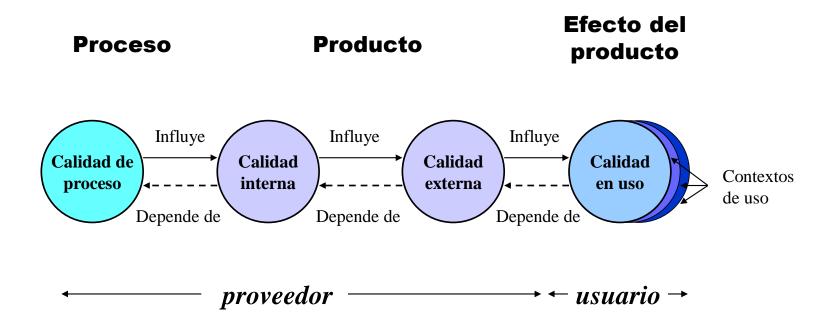
- El objetivo no es necesariamente alcanzar una calidad perfecta, sino la necesaria y suficiente para cada contexto de uso a la hora de la entrega y del uso por parte de los usuarios.
- Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (requisitos).

Cuando no hay Calidad

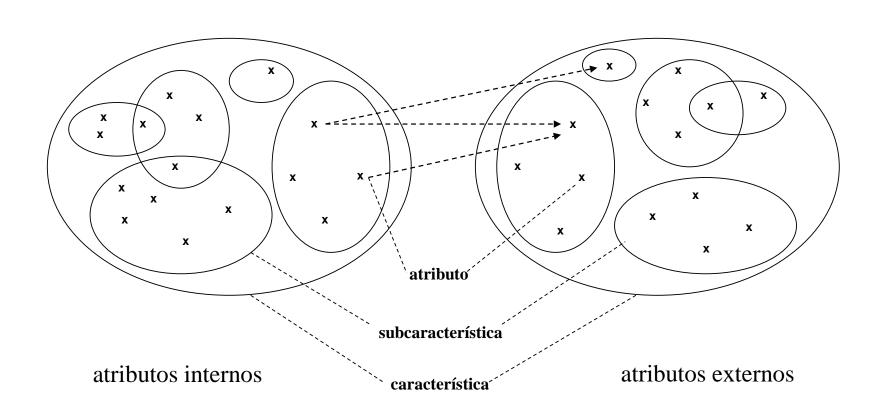
- ❖ Programas que no hacen exactamente lo que se espera □
- Proyectos que no terminan nunca
- ❖ Sistemas informáticos que no se utilizan por la dificultad de su manejo □
- ❖ Productos software que son imposibles de mantener cuando desaparece la persona o personas que lo desarrollaron □
- Software poco seguro

Diferentes aspectos de la calidad

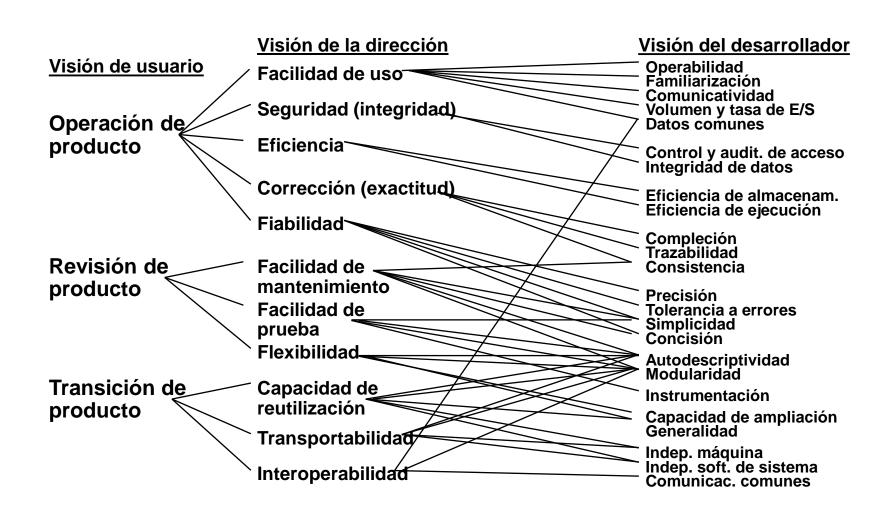
- Interna: medible a partir de las características intrínsecas, como el código fuente
- Externa: medible en el comportamiento del producto, como en una prueba
- En uso: durante la utilización efectiva por parte del usuario



Características, subcaracterísticas y atributos de calidad



Modelo de McCall et al. (1977)



- ❖ISO/IEC 9126:2001 □
- ❖ISO/IEC 14598 □
- Familia de normas ISO 25000

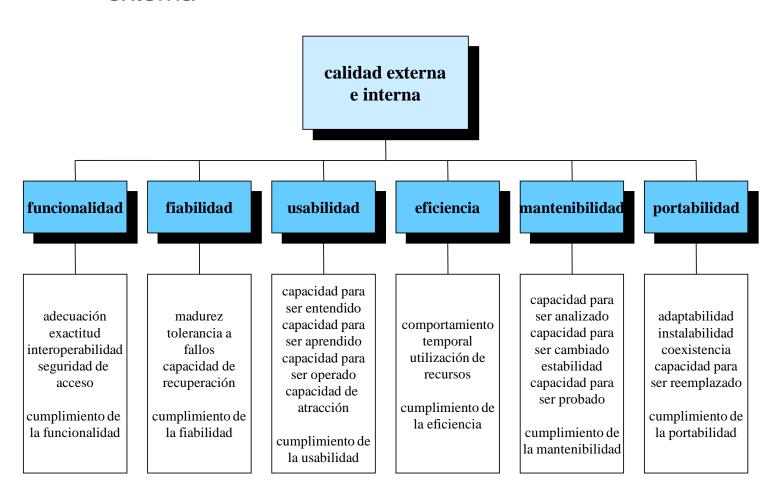
ISO/IEC 9126: Tecnologías de la Información – Calidad de los productos software.

- Parte 1: Modelo de Calidad
- Parte 2: Métricas Externas
- Parte 3: Métricas Internas
- Parte 4: Métricas de Calidad en Uso

Ejemplos de uso:

- Validar la compleción de una definición de requisitos
- Identificar requisitos software
- Identificar objetivos para el diseño software
- Identificar requisitos para las pruebas del software
- Identificar requisitos para el aseguramiento de la calidad
- Identificar criterios de aceptación para un producto software terminado

Modelo de calidad para calidad interna y externa



Funcionalidad

Adecuación

Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Exactitud

Capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.

Interoperabilidad

Capacidad del producto software para interactuar con uno o más sistemas especificados.

Seguridad de acceso

Capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados

Cumplimiento funcional

Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.

Fiabilidad

Madurez

Capacidad del producto software para evitar fallar como resultado de fallos en el software.

Tolerancia a fallos

Capacidad del software para mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.

Capacidad de recuperación

Capacidad del producto software para reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo.

Cumplimiento de la fiabilidad

Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con al fiabilidad.

Usabilidad

Capacidad para ser entendido

Capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.

Capacidad para ser aprendido

Capacidad del producto software que permite al usuario aprender sobre su aplicación.

Capacidad para ser operado

Capacidad del producto software que permite al usuario operarlo y controlarlo.

Capacidad de atracción

Capacidad del producto software para ser atractivo al usuario.

Cumplimiento de la usabilidad

Capacidad del producto software para adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.

Eficiencia

Comportamiento temporal

Capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas.

Utilización de recursos

Capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

Cumplimiento de la eficiencia

Capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia.

Mantenibilidad

Capacidad para ser analizado

Es la capacidad del producto software para serle diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.

Capacidad para ser cambiado

Capacidad del producto software que permite que una determinada modificación sea implementada.

Estabilidad

Capacidad del producto software para evitar efectos inesperados debidos a modificaciones del software.

Capacidad para ser probado

Capacidad del producto software que permite que el software modificado sea validado.

Cumplimiento de la mantenibilidad

Capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

Portabilidad

Adaptabilidad

Capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.

Instalabilidad

Capacidad del producto software para ser instalado en un entorno especificado.

Coexistencia

Capacidad del producto software para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes.

Capacidad para reemplazar

Capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.

Cumplimiento de la portabilidad

Capacidad del producto software para adherirse a normas o

Modelo de calidad para calidad en uso



Efectividad

Capacidad del producto software para permitir a los usuarios alcanzar objetivos especificados con exactitud y completitud, en un contexto de uso especificado.

Productividad

Capacidad del producto software para permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada, en un contexto de uso especificado.

Seguridad física

Capacidad del producto software para alcanzar niveles aceptables del riesgo de hacer daño a personas, al negocio, al software, a las propiedades o al medio ambiente en un contexto de uso especificado.

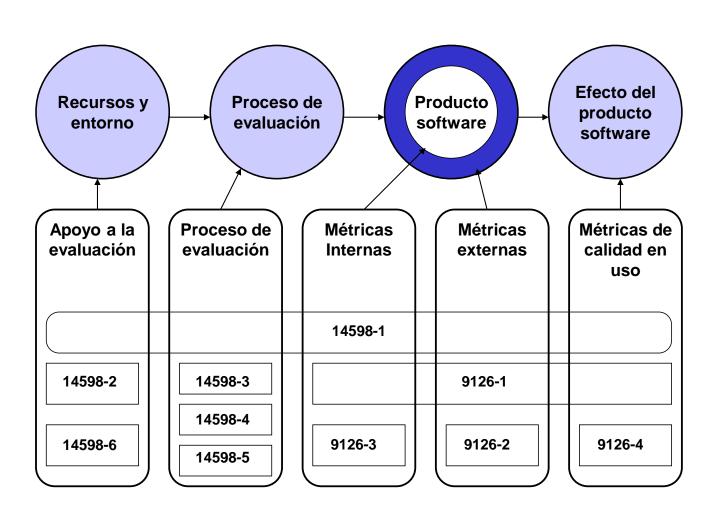
Satisfacción

Capacidad del producto software para satisfacer a los usuarios en un contexto de uso especificado.

- ❖ Detectar los defectos en el producto software y proceder a su eliminación antes de la entrega, lo que supone un ahorro de costes en la fase de mantenimiento posterior.
- Evaluar y controlar el rendimiento del producto software desarrollado, asegurando que podrá generar los resultados teniendo en cuenta las restricciones de tiempo y recursos establecidas.

- Asegurar que el producto software desarrollado respeta los niveles necesarios para las características de seguridad (confidencialidad, integridad, autenticidad, no-repudio, etc.).
- Comprobar que el producto desarrollado podrá ser puesto en producción sin poner en compromiso el resto de sistemas y manteniendo la compatibilidad con las interfaces necesarias.

Evaluación del producto software: ISO 14598

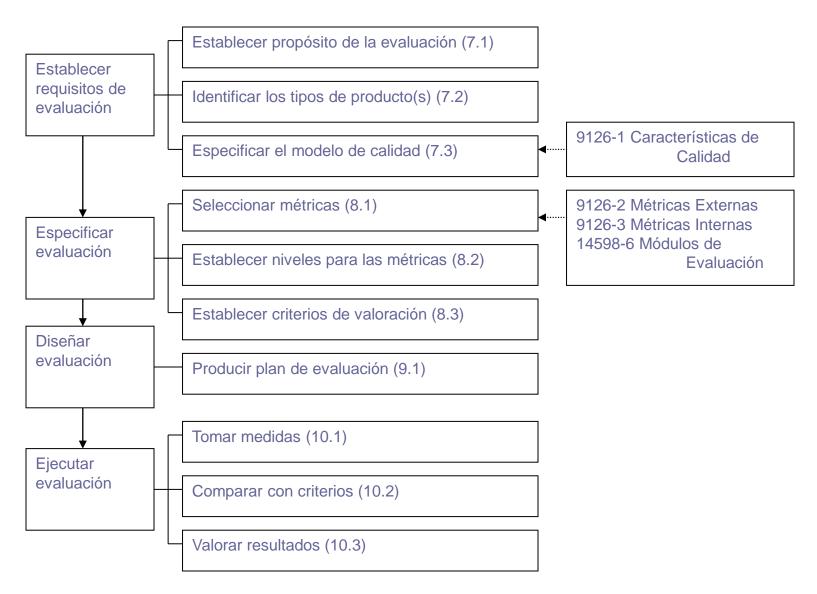


La norma UNE 71048: Tecnología de la Información – Evaluación del Producto Software (Soporte Lógico):

- -- Parte 1: Visión general
- -- Parte 2: Planificación y gestión
- -- Parte 3: El proceso para desarrolladores
- -- Parte 4: El proceso para adquisidores
- -- Parte 5: El proceso para evaluadores
- -- Parte 6: Documentación de los módulos de evaluación

CALIDAD DE PRODUCTO

Proceso de evaluación



Establecer el propósito de la evaluación

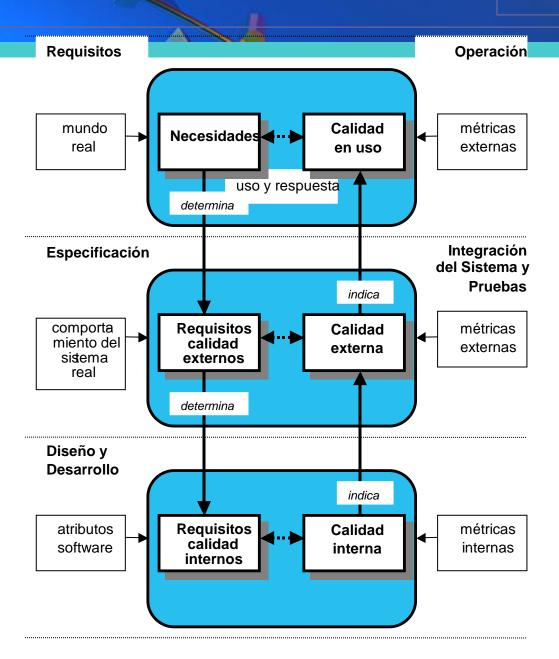
Productos intermedios:

- decidir sobre la aceptación de un producto intermedio de un subcontratista;
- decidir cuando un proceso está completo y cuando remitir los productos al siguiente proceso;
- predecir o estimar la calidad del producto final;
- recoger información con objeto de controlar y gestionar el proceso.

Producto final:

- decidir sobre la aceptación del producto;
- decidir cuando publicar el producto;
- comparar el producto con otros productos competitivos;
- seleccionar un producto entre productos alternativos;
- valorar tanto el aspecto positivo como negativo cuando está en uso;
- decidir cuando mejorar o reemplazar un producto.

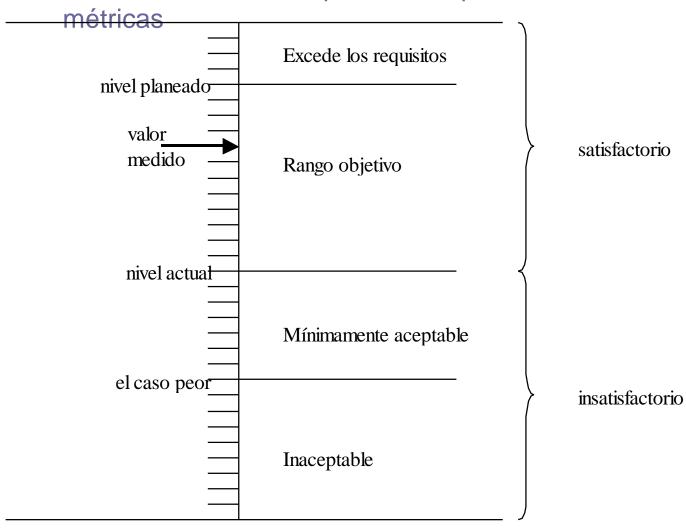
CALIDAD DE PRODUCTO



Identificar
los tipos de
producto(s)
a ser
evaluados

CALIDAD DE PRODUCTO

Establecer niveles de puntuación para las

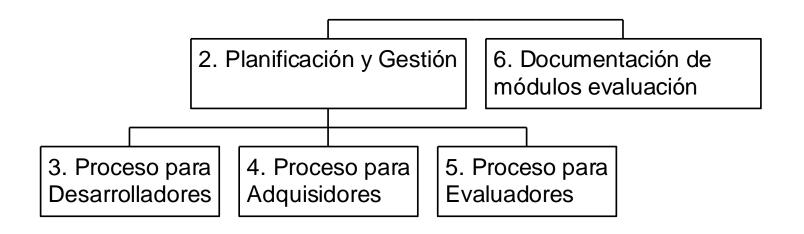


escala de medición

niveles de puntuación

Producir un plan de evaluación

El plan de evaluación describe los métodos de evaluación y el programa de acciones del evaluador (UNE 71048-3, UNE 71048-4 o UNE 71048-5). Debe ser consistente con el plan de mediciones (UNE 71048-2).



CALIDAD DE PRODUCTO

	Modelo de Calidad 2501n	
Requisitos de Calidad 2503n	Gestión de Calidad 2500n	Evaluación de Calidad 2504n
	Medición de Calidad 2502n	

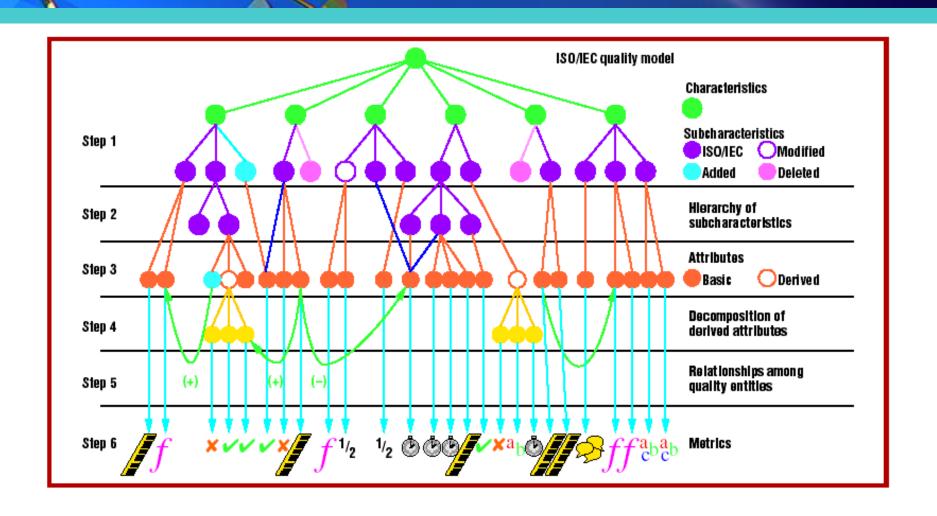
CALIDAD DEL PRODUCTO

FUNCIONALIDAD	Completitud Corrección Idoneidad	SEGURIDAD	{	Confidencialidad Integridad No repudio
RENDIMIENTO	Comportamiento en el tiempo		l	Autenticidad Responsabilidad
RENDIMIENTO	Utilización de Recursos			Modularidad Reusabilidad
USABILIDAD	Inteligibilidad Aprendizaje Operabilidad Protección a errores de usuario	MANTENIBILIDAD		Analizabilidad Cambiabilidad Capacidad de ser probado
	Atractividad Accesibilidad	PORTABILIDAD	{	Adaptabilidad Facilidad de instalación
FIABILIDAD	Madurez Disponibilidad Tolerancia a fallos			Intercambiabilidad
	Capacidad de recuperación	COMPATIBILIDAD	$\left\{ \right.$	Coexistencia Interoperabilidad

Franch y Carvallo (2003)

- 0.- Definir el dominio
- 1.- Determinar subcaracterísticas de calidad
- 2.- Definir una jerarquía de subcaracterísticas
- 3.- Descomponer subcaracterísticas en atributos
- 4.- Descomponer atributos derivados en atributos básicos
- 5.- Establecer relaciones entre entidades de calidad
- 6.- Determinar métricas para los atributos

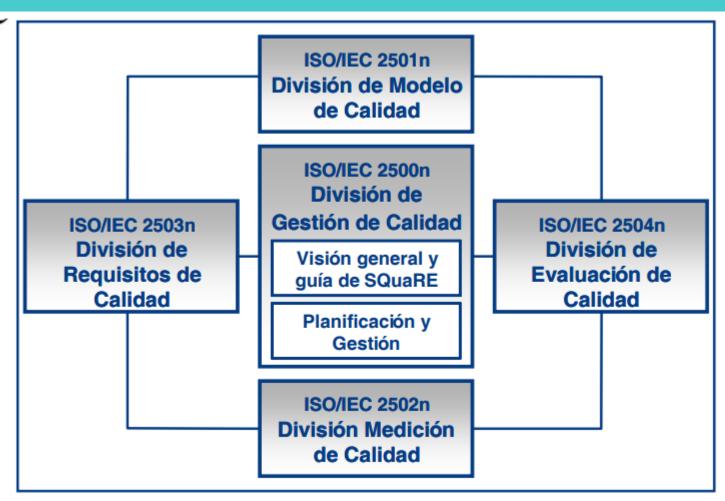
ESEMPLOS



Motivos para evaluar productos Según ISO/IEC 25000

- Diferenciarse de los competidores, asegurando tiempos de entrega y reducción de fallos en el producto tras su implantación en producción.
- Establecer acuerdos en el ámbito del servicio, definiéndose determinados parámetros de calidad que el producto debe cumplir antes de ser entregado.

- Nace por las inconsistencias entre ISO 9126 e ISO 14598. □
- El objetivo es aglutinar bajo una misma familia el modelo de calidad y el proceso de evaluación.



Zubrow, D. (2004). Measuring Software Product Quality: the ISO 25000 Series and CMMI. SEI.

Calidad de Producto Software - ISO/IEC 25000

54







Procesos de Gestión de Calidad del Software



Contenido



La Gestión de la Calidad

Establece un marco de proceso y estándares de organización que conducirán a obtener software de mejor calidad.

Implica la aplicación de procesos específicos de calidady la verificación de que continúen dichos procesos planeados

La Gestión de la Calidad

- Establecer un plan de calidad para un proyecto.
 - El plan de calidad debe establecer metas de calidad para el proyecto y definir cuáles procesos y estándares se usarán.

Gestión de la Calidad

- Proporciona una comprobación independiente sobre el proceso de desarrollo de software.
- Verifica los entregables del proyecto para garantizar que sean consistentes con los estándares y metas de la organización

Gestión de la calidad

Asegurar que los productos de trabajo y la ejecución de los procesos estén en conformidad con los planes, procedimientos y estándares establecidos

Aseguramiento de la Calidad (QA)

Es la definición de procesos y estándares que deben conducir a la obtención de productos de alta calidad.

Representa simplemente la definición de procedimientos, procesos y estándares cuyo objetivo es asegurar el logro de la calidad del software.

Conformidad del Proceso y del Producto

- Busca asegurar que los productos producidos cumplan con las características de calidad preestablecidas.
- Busca asegurar que los procesos planificados sean implementados.

Conformidad del Proceso y del Producto

- Cuando no-conformidades son identificadas, ellas deben ser tratadas y resueltas en el proyecto.
- En caso de que no sean resueltas en el proyecto, deben ser escalonadas para el nivel adecuado de gerencia

Evaluar Objetivamente

- La objetividad es crítica para el éxito del proyecto.
- La objetividad se consigue con:
 - El evaluador independiente del proyecto (externo al proyecto) → Grupo de Aseguramiento de la calidad.
 - La utilización de un conjunto de criterios de evaluación→ disminuye la subjetividad y el vicio del evaluador.

Plan de Calidad

- 1. Introducción del producto
- 2. Planes del producto
- 3. Descripciones de procesos
- 4. Metas de calidad
- 5. Riesgos y gestión del riesgo

Watts Humphrey (1989)

Plan de Calidad

- Se desarrollan como parte del proceso de planeación general del proyecto.
- Difieren en detalle dependiendo del tamaño y tipo de sistema, contexto del proyecto y necesidades del cliente y la organización.
- Tratar de mantenerlos tan simples como sea posible

Preguntas sobre el sistema

- ❖ ¿En el proceso de desarrollo se siguieron los estándares de programación y documentación?
- ¿El software se verificó de manera adecuada?
- ¿El software es suficientemente confiable para utilizarse?

Preguntas sobre el sistema

- ❖ ¿El rendimiento del software es aceptable para su uso normal?
- ❖ ¿El software está bien estructurado y es comprensible?

Tipos de estándares

Estándares del Producto

- Se aplican al producto software a desarrollar.
- Incluyen estándares de documentos, documentación y estándares de codificación.

Estándares del Proceso

- Establecen procesos que deben seguirse durante el desarrollo del software.
- Incluyen definiciones de especificación, procesos de diseño y validación, etc.

Tipos de estándares

Estándares de Producto	Estándares de Proceso
Formato de revisión de diseño	Realizar revisión de diseño
Estructura de documento de	Enviar nuevo código para
requerimientos	construcción de sistema
Formato de encabezado por	
método	Proceso de liberación de versión
método	Proceso de liberación de versión Proceso de aprobacióndel Plan
método Estilo de programación Java	
	Proceso de aprobacióndel Plan

Atributos de calidad

- Protección
- Seguridad
- Fiabilidad
- Flexibilidad
- Robustez
- Comprensibilidad
- Adaptabilidad
- Modularidad

- Complejidad
- Portabilidad
- Usabilidad
- Reusabilidad
- Eficiencia
- Facilidad para que el usuario aprenda a utilizarlo

Buenas prácticas

- Evaluada objetivamente la adherencia de los productos a los estándares, procedimientos y requisitos aplicables,
 - Antes de que los productos sean entregados y
 - En hitos predefinidos a lo largo del ciclo de vida del proyecto.

Buenas prácticas

- Evaluada objetivamente la adherencia de los procesos ejecutados a las descripciones de proceso, estándares y procedimientos.
 - Realizar evaluaciones, a lo largo del ciclo de vida, por personas fuera del contexto del proyecto y basadas en criterios de adherencia a los procesos.

Buenas Practicas

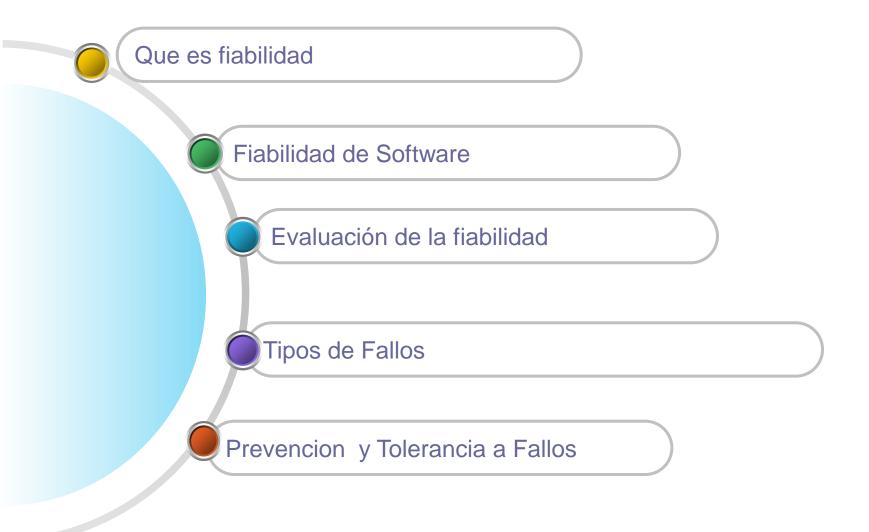
- ❖ Identificar, registrar y comunicar los problemas y las no-conformidades.
 - Registrar las no-conformidades identificadas.
 - Comunicar no-conformidades a los responsables por los productos y/o procesos aplicables.
 - Se debe buscar identificar el origen de los problemas para analizar la necesidad de alteración de procesos, estándares y procedimientos.



Tema 4. Fiabilidad



Contenido



¿Qué es la fiabilidad?

Fiabilidad:

Permanencia de la calidad de los productos (o servicios) a lo largo del tiempo.

Capacidad de desarrollar adecuadamente su labor a lo largo del tiempo.

Definición AFNOR X 06-501

"Fiabilidad es la característica de un dispositivo expresada por la probabilidad de que un dispositivo cumpla una función requerida en las condiciones de utilización y para un período de tiempo determinado".

Probabilidad: es la relación número de casos favorables número de casos posibles asociada a un tiempo t .

Se denomina R (t) = P (cumplir una misión) = P (buen funcionamiento) R traducción del inglés *Reliability*

La Fiabilidad intenta garantizar que el producto permanecerá en buenas condiciones durante un periodo razonable de tiempo

Según la definiciones se tiene que pensar muy claramente qué significa

- Funcionamiento satisfactorio
- Tiempo de funcionamiento (Misión)
- Condiciones de funcionamiento

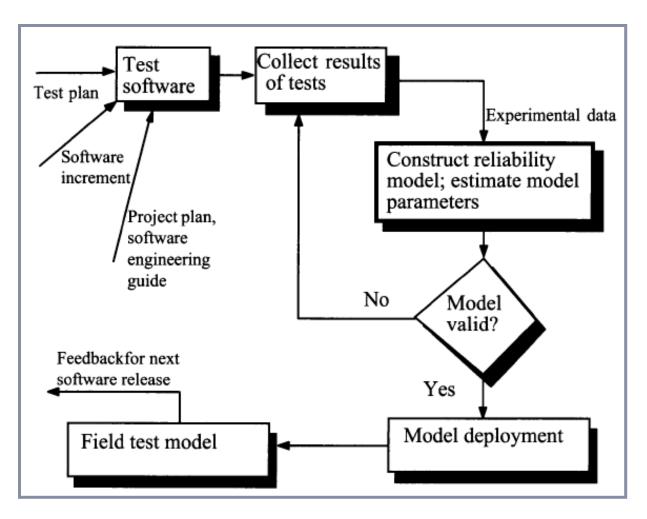
Necesidad de fiabilidad

- Desde un punto de vista puramente económico, es deseable una alta fiabilidad para reducir los costos totales del producto.
- El hecho de que en algunos sistemas militares el costo anual de mantenimiento sea diez veces el costo original del mismo, pone de manifiesto esta necesidad. (ciclo vida)
- También hay que considerar el aspecto de seguridad (el fallo de un sistema ABS en un automóvil puede ser catastrófico).
- Existen otro aspectos como retrasos de horarios, incomodidades, insatisfacción del cliente y pérdida de prestigio del fabricante.
- Cada vez son más las empresas y organismos que en sus contrataciones exigen ciertas normas de fiabilidad (MIL HDBK 217 en USA...)

Fiabilidad de software

❖ Fiabilidad, capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas.

Evaluacion de Fiabilidad



El Rol de la Fiabilidad en Ingenieria de Software

La confiabilidad es probablemente la más importante de las características inherentes en el concepto de "calidad del software". La confiabilidad del software se refiere a qué tan bien funciona el sofware para cumplir con los requisitos del cliente.

j.Musa y otros 1990

Error, Fault and Failure



- Error: human action that results in software containing a fault
- Fault: a cause for an internal error (failure)
- Failure: any observable divergence of software behavior in execution from user needs
- * Failure intensity: the number of failures per time unit

Fallos de funcionamiento

- Los fallos de funcionamiento de un sistema pueden tener su origen en
 - Una especificación inadecuada
 - Errores de diseño del software
 - Averías en el hardware
 - Interferencias transitorias o permanentes en las comunicaciones u Nos centraremos en el estudio de los errores de software
- Nos centraremos en el estudio de los errores de software

- La fiabilidad (reliability) de un sistema es una medida de su conformidad con una especificación autorizada de su comportamiento
- Una avería (failure) es una desviación del comportamiento de un sistema respecto de su especificación
- Las averías se manifiestan en el comportamiento externo del sistema, pero son el resultado de errores (errors) internos
- Las causas mecánicas o algorítmicas de los errores se llaman fallos (faults)

 Los fallos pueden ser consecuencia de averías en los componentes del sistema (que son también sistemas)



Tipos de fallos

Fallos transitorios

- desaparecen solos al cabo de un tiempo
- ejemplo: interferencias en comunicaciones

Fallos permanentes

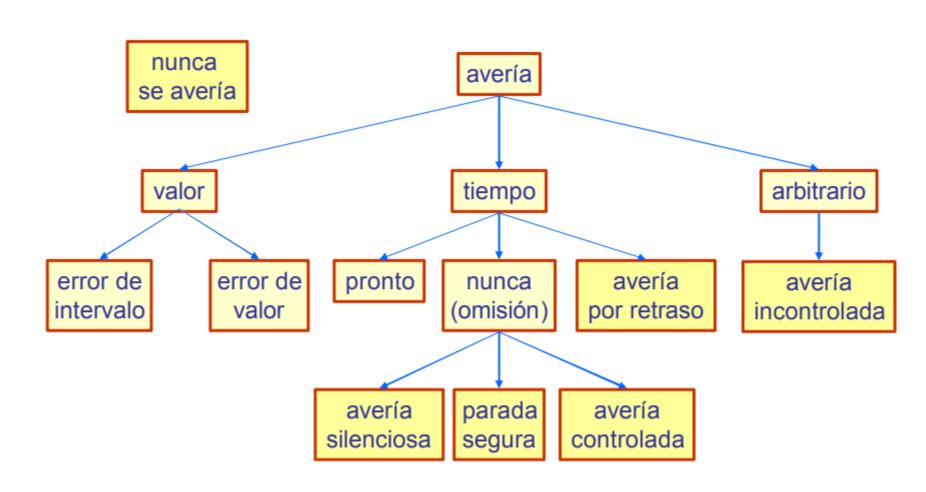
- permanecen hasta que se reparan
- ejemplo: roturas de hardware, errores de diseño de software

Fallos intermitentes

- fallos transitorios que ocurren de vez en cuando
- ejemplo: calentamiento de un componente de hardware

Debe impedirse que los fallos de todos estos tipos causen averías

Tipos de avería (failure modes)



Prevención y tolerancia de fallos

- Hay dos formas de aumentar la fiabilidad de un sistema:
 - Prevención de fallos
 - Se trata de evitar que se introduzcan fallos en el sistema antes de que entre en funcionamiento
 - Tolerancia de fallos
 - Se trata de conseguir que el sistema continúe funcionando aunque se produzcan fallos
- En ambos casos el objetivo es desarrollar sistemas con tipos de averías bien definidos

Prevención de fallos

Se realiza en dos etapas:

- Evitación de fallos
 - Se trata de impedir que se introduzcan fallos durante la construcción del sistema
- Eliminación de fallos
 - Consiste en encontrar y eliminar los fallos que se producen en el sistema una vez construido

Técnicas de evitación de fallos

Hardware

- Utilización de componentes fiables
- Técnicas rigurosas de montaje de subsistemas
- Apantallamiento de hardware

Software

- Especificación de requisitos rigurosa o formal
- Métodos de diseño comprobados
- Lenguajes con abstracción de datos y modularidad
- Utilización de entornos de desarrollo con computador (CASE) adecuados para gestionar los componentes

Técnicas de eliminación de fallos

Comprobaciones

- Revisiones de diseño
- Verificación de programas
- Inspección de código

Pruebas (tests)

- Son necesarias, pero tienen problemas:
 - no pueden ser nunca exhaustivas
 - sólo sirven para mostrar que hay errores, no que no los hay
 - a menudo es imposible reproducir las condiciones reales
 - los errores de especificación no se detectan

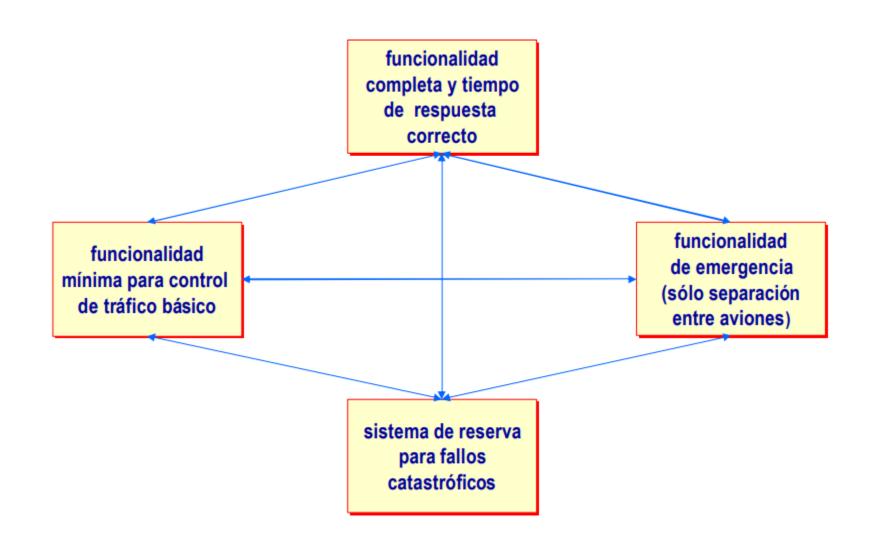
Limitaciones de la prevención de fallos

- Los componentes de hardware fallan, a pesar de las técnicas de prevención
 - La prevención es insuficiente si
 - la frecuencia o la duración de las reparaciones es inaceptable
 - no se puede detener el sistema para efectuar operaciones de mantenimiento
- Ejemplo: naves espaciales no tripuladas
- La alternativa es utilizar técnicas de tolerancia de fallos

Grados de tolerancia de fallos

- Tolerancia completa (fail operational)
 - El sistema sigue funcionando, al menos durante un tiempo, sin perder funcionalidad ni prestaciones
- Degradación aceptable (fail soft, graceful degradation)
 - El sistema sigue funcionando con una pérdida parcial de funcionalidad o prestaciones hasta la reparación del fallo
- Parada segura (fail safe)
 - El sistema se detiene en un estado que asegura la integridad del entorno hasta que se repare el fallo El grado de tolerancia de fallos necesario depende de la aplicación

Ejemplo: control de tráfico aéreo



Redundancia

- La tolerancia de fallos se basa en la redundancia
- Se utilizan componentes adicionales para detectar los fallos y recuperar el comportamiento correcto
- Esto aumenta la complejidad del sistema y puede introducir fallos adicionales
- Es mejor separar los componentes tolerantes del resto del sistema

Redundancia en hardware

Redundancia estática

- Los componentes redundantes están siempre activos
- Se utilizan para enmascarar los fallos
- Ejemplo:
 - Redundancia modular triple (ó N), TMR/NMR

Redundancia dinámica

- Los componentes redundantes se activan cuando se detecta un fallo
- Se basa en la detección y posterior recuperación de los fallos
- Ejemplos:
 - sumas de comprobación » bits de paridad

Tolerancia de fallos de software

- Técnicas para detectar y corregir errores de diseño
- Redundancia estática
 - Programación con N versiones
- Redundancia dinámica
 - Dos etapas: detección y recuperación de fallos
 - Bloques de recuperación
 - Proporcionan recuperación hacia atrás
 - Excepciones
 - Proporcionan recuperación hacia adelante

Programación con N versiones

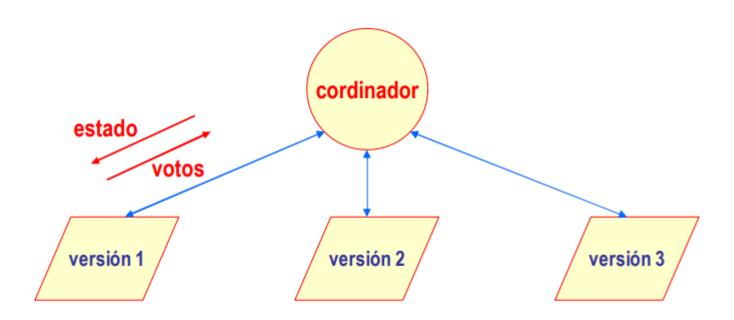
Diversidad de diseño

- N programas desarrollados independientemente con la misma especificación
- sin interacciones entre los equipos de desarrollo

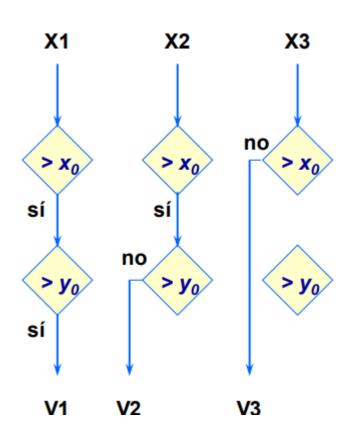
Ejecución concurrente

- proceso coordinador (driver)
 - » intercambia datos con los procesos que ejecutan las versiones
- todos los programas tienen las mismas entradas
- las salidas se comparan
- si hay discrepancia se realiza una votación

Programación con N versiones



Comparación consistente



- La comparación de valores reales o texto no es exacta
- •Cada versión produce un resultado correcto, pero diferente de las otras
- •No se arregla comparando con x0+D, y0+D

Problemas

- La correcta aplicación de este método depende de:
 - Especificación inicial
 - Un error de especificación aparece en todas las versiones
 - Desarrollo independiente
 - No debe haber interacción entre los equipos
 - No está claro que distintos programadores cometan errores independientes
 - Presupuesto suficiente
 - Los costes de desarrollo se multiplican u ¿sería mejor emplearlos en mejorar una versión única?
 - El mantenimiento es también más costoso
- Se ha utilizado en sistemas de aviónica críticos

Redundancia dinámica en software

Cuatro etapas:

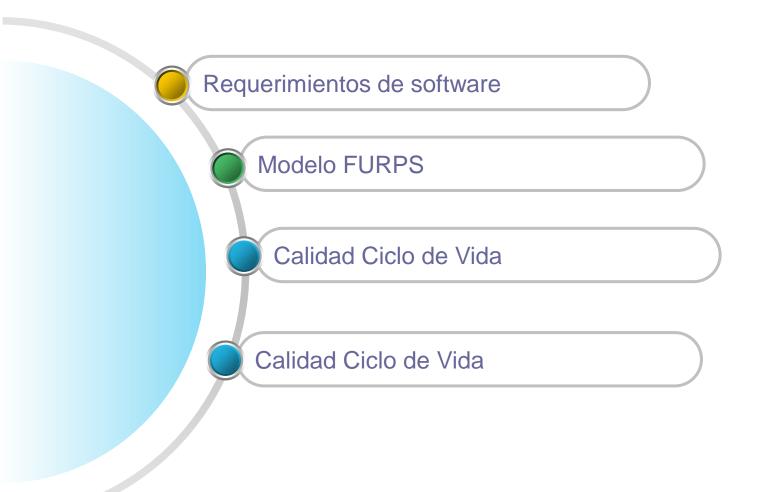
- Detección de errores
 no se puede hacer nada hasta que se detecta un error
- 2. Evaluación y confinamiento de los daños » diagnosis: averiguar hasta dónde ha llegado la información errónea
- 3. Recuperación de errores llevar el sistema a un estado correcto, desde el que pueda seguir funcionando (tal vez con funcionalidad parcial)
- 4. Reparación de fallos Aunque el sistema funcione, el fallo puede persistir y hay que repararlo



Requerimientos de Calidad de Software



Contenido



Requerimientos de Software

- La comprensión de los requisitos de un problema está entre las tareas más difíciles que enfrenta el Ingeniería de Software.
- La Ingeniería de Requisito ayuda a los Ingenieros de Software a entender mejor el problema.
- ❖ Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software.

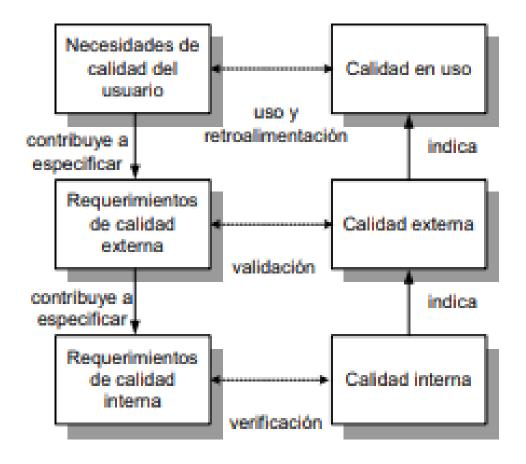
Requerimientos de Calidad de Software

El modelo FURPS+

```
Functional;
Usability;
Modelo FURPS+ Reliability;
Performance;
Supportability
```

❖ El modelo FURPS incluye, además de los factores de calidad y los atributos, restricciones de diseño y requerimientos de implementación, físicos y de interfaz.

Calidad en el ciclo de vida del software. Tomado de ISO/IEC 9126



La traducción de los requisitos de calidad a nivel del usuario hacia la calidad externa e interna representan un problema

PASOS PARA LA CALIDAD DE PRODUCTO SEGÚN LA NORMA ISO/IEC 9126

- 1. identificación de requerimientos de calidad
- 2. especificación de la evaluación,
- 3. diseño de la evaluación,
- 4. ejecución de la evaluación
- 5. retroalimentación a la organización

El primer paso debe realizarse durante el Análisis de los Requerimientos

Identificación de Requerimientos de Calidad

- ❖ Permite determinar los pesos a ser utilizados en el modelo de calidad y que debe reflejar las necesidades de calidad del usuario para cada una de las características y sub características.
- ❖ Los pesos representan la valoración comparada entre las distintas características y sub características y para ello se puede utilizar una calificación relativa de alto / medio / bajo o una calificación basada en valores que puede ir entre 1 y 9.

La especificación de la evaluación

- ❖ Permite identificar los valores deseables de las métricas a utilizar posteriormente en el desarrollo y en la evaluación del producto.
- Estos valores deben orientarse principalmente a cubrir las necesidades del usuario.
- La definición de valores deseables depende directamente de cada atributo del producto.

El diseño de la evaluación

comprende la preparación de un plan de medición conteniendo los entregables sobre los cuales se hará el proceso de medición y las métricas que se aplicarán.

DReC

Una técnica para la determinación de los requerimientos de calidad de un producto software basada principalmente en el punto de vista de usuarios y el punto de vista de desarrolladores de una manera complementaria

La definición implica

- ❖ la determinación de requerimientos de calidad es un proceso de fijación de valores que serán tomados inicialmente para la planificación de la calidad y posteriormente como referencia para la evaluación del producto software.

- el desarrollador es un actor que contrapesará la opinión del usuario, pero debe subordinar –finalmente- su opinión a la del usuario si no existe un consenso sobre los valores;
- ❖ DReC se orienta principalmente a usuarios finales, por lo que la manera de relacionarse a él, será en términos lo menos técnico posible pero con la claridad necesaria para determinar adecuadamente los valores;

Objetivos drREC

- las características de calidad interna y externa que son relevantes en la desarrollo de software
- los niveles de calidad de cada característica y sub-característica
- los niveles de calidad de los atributos (valores deseables de las métricas) del producto a desarrollar

Descripcion DREC

Paso 1: Seleccionar componentes.

- objetivo es seleccionar un conjunto de componentes a los cuales se les aplicará el resto de la técnica.
- un producto software tiene diversos componentes cuyas necesidades de calidad son diferentes, dependiendo de la función que realizan dentro del producto final.
- resultado del paso es una lista de componentes seleccionados, donde cada componente puede ser distinguible por usuarios y desarrolladores

Subpasos de seleccionar componentes

Paso 1a. Descomponer el producto software en componentes

 utilizar una estructura de descomposición del trabajo orientada al producto también conocido como WBS (del inglés Work Breakdown Structure).

Paso 1b. Seleccionar los componentes relevantes

 Componentes considerados los más importantes y/o críticos para la solución (sistema software) que se va a desarrollar.

Descripcion DREC

- Paso 2.Definir los pesos del modelo de calidad (características y subcaracteristicas)
 - objetivo es la determinación de los valores a ser usados en el modelo obtenidos en el paso
 1
 - Cada participante contestará el cuestionario de modo que plasme su percepción sobre la importancia –comparada- de las características y sub características.
 - Los participantes son usuarios y desarrolladores

Subpasos de Definir los pesos del modelo de calidad

- ❖ Paso 2a. Completar la hoja inicial (DReC00) marcando con una "x" de acuerdo a cada línea que describe una característica o sub característica.
 - La hoja es completada por todas las personas convocadas: usuarios y desarrolladores.
- Paso 2b. Consolidar la información de todos los participantes
 - Debe ser rápido y con resultados confiables.

Paso 2c. Revisar los resultados obtenidos y discutir sobre las respuestas

- cuya variación sea significativa hasta encontrar un consenso entre todos los participantes de la sesión, la participación será mediante un director de debate.
- Se podrá utilizar adicionalmente las hojas "DReC11" y "DReC12" siempre que se cuente con datos históricos.

Descripción DREC

Paso 3: Definir los niveles de calidad esperado en los atributos del producto

- objetivo es la determinación de los valores a ser usados como nivel de referencia para las métricas en la evaluación
- La razón es que la calidad de un producto es finalmente evaluada por el usuario cada vez que utiliza el software
- El resultado de este paso es un hoja con los niveles de calidad en cada atributo del producto

Subpasos de Definir los niveles de calidad

- Paso 3a. Completar la hoja "DReC01" y "DReC02"
 - marcando con una "x" de acuerdo a cada línea que describe un atributo.
 - La hoja es completada por todas las personas convocadas: usuarios y desarrolladores
- Paso 3b. Consolidar la información de todos los participantes



Paso 3c. Revisar los resultados obtenidos y discutir sobre las respuestas

 cuya variación sea significativa hasta encontrar un consenso entre todos los participantes de la sesión, la participación será mediante un director de debate

APLICACIÓN DE LA TÉCNICA

DReC se debe aplicar en dos etapas principales:

paso 1.

 el equipo de desarrollo es el encargado de hacer la descomposición y la selección de componentes tomando en cuenta las necesidades del usuario,

Paso 2 y 3 en conjunto

 Para el paso 2 y 3 se convoca a un conjunto de personas entre usuarios y desarrolladores de modo que realicen las actividades indicadas para cada componente

Resumen DREC



Ejemplo de Aplicación

Calidad Externa e			
Interna			
Característica	Peso %	Subcaracterística	Peso %
Funcionalidad	21	Aplicabilidad	39
		Precisión	36
		Interoperatibilidad	7
		Seguridad	8
		Conformidad de funcionalidad	10
Fiabilidad	22	Madurez (hardware/software/datos)	40
		Tolerancia a fallos	36
		Recuperabilidad (datos, proceso, tecnología)	14
		Conformidad de fiabilidad	10
Usabilidad	15	Entendibilidad	29
		Facilidad de aprendizaje	23
		Operabilidad	27
		Atractividad	11
		Conformidad de usabilidad	10
Eficiencia	18	Comportamiento en el tiempo	41
		Utilización de recursos	35
		Conformidad de eficiencia	24
Facilidad de	16	Analizabilidad	21
Mantenimiento			
		Cambiabilidad	23
		Estabilidad	21
		Testeabilidad	27
		Conformidad de facilidad de mantenimiento	8
Portabilidad	8	Adaptabilidad	25
		Instabilidad	25
		Co existencia	25
		Reemplazabilidad	13
		Conformidad de portabilidad	13