



# INGENIERÍA DEL SOFTWARE II

## Tema 3

### *Calidad del Software*

*Univ. Cantabria – Fac. de Ciencias*

*Francisco Ruiz, Mario Piattini (UCLM)*



### Objetivos del Tema

- Comprender el concepto de **calidad**.
- Conocer las características generales de la **calidad del software**, según normas internacionales.
- Conocer los diferentes aspectos relacionados con la calidad de los **procesos y productos software**.
- Comprender la importancia de la **medición** en la búsqueda de la calidad.
- Conocer las principales **técnicas y métodos** para mejorar la calidad, incluidas la **verificación y validación**.



## Contenido

- Introducción
  - Concepto de Calidad
  - Perspectivas de la Calidad
  - ISO 9000:2000
- Calidad de Sistemas de Información
- Calidad del Software
  - Factores
- Calidad de Producto
  - ISO 9126
  - Calidad de Datos
- Calidad de Proceso
  - ISO 90003
- Evaluación y Mejora de Procesos
  - CMMI
  - ISO 15504
  - PSP Y TSP
- Medición del Software
  - Proceso
  - GQM
  - Medidas
- Gestión de la Calidad
  - Aseguramiento
- Verificación y Validación
  - Revisiones
  - Auditorias



## Bibliografía

- Básica
  - SWEBOK - Guide to the Software Engineering Body of Knowledge, 2004 Version.
    - Cap. 11.
  - Piattini et al. (2006): Calidad de Sistemas Informáticos. Ra-Ma.
    - Caps. 4-5 y 8.
- Complementaria
  - Piattini et al. (2006): Calidad de Sistemas Informáticos. Ra-Ma.
    - Caps. 1, 3, 9-10.
  - Sommerville (2005): Ingeniería del Software. 7ª edición. Addison-Wesley.
    - Caps. 27 y 28.
  - Pressman (2005): Ingeniería del Software: Un Enfoque Práctico. 6ª Edición. McGraw-Hill.
    - Cap. 26.



## Agradecimientos

---

- Este material ha sido preparado con la colaboración de:
  - Mario Piattini (Universidad de Castilla-La Mancha).
  - Félix García (Universidad de Castilla-La Mancha).
  - Ismael Caballero (Universidad de Castilla-La Mancha).



## 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***



## Introducción

- Origen del Interés Actual por la Calidad:
  - En un mercado competitivo no es suficiente con producir y distribuir masivamente productos o servicios.
  - La **Calidad** se convierte en un objetivo fundamental junto con los otros dos parámetros clásicos de **Dinero** y **Tiempo**.
  - Gran Importancia y atención a las **Necesidades del Cliente**
    - Investigación de Mercados, Especificaciones



## Introducción – Concepto de Calidad

- **Definición coloquial:**
  - En la **Vida Cotidiana** “la calidad representa las propiedades inherentes a un objeto que permiten apreciarlo como mejor, igual o peor que otros objetos de su especie”.
  - Es sinónimo de bondad, excelencia o superioridad”.
  - ¿Esta idea de calidad es aplicable de manera formal en una empresa?
    - Crosby → “La calidad puede no ser lo que piensas”
  - **DRAE:**
    - *Propiedad o conjunto de propiedades que, inherentes a una cosa, permiten apreciarla como igual, mejor o peor que las restantes de su especie.*
    - *En sentido absoluto, buena calidad, superioridad o excelencia.*



## Introducción – Concepto de Calidad

- **Definición Profesional:**

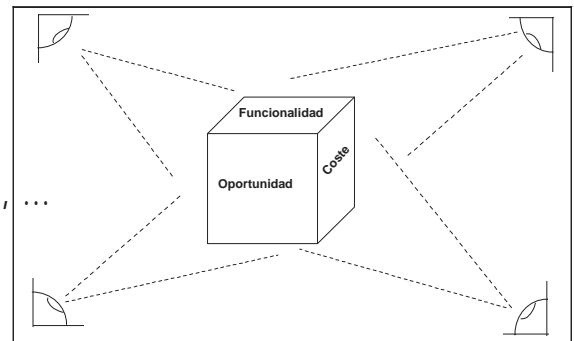
- Totalidad de las **características y aspectos de un producto o servicio** en los que se basa su aptitud para **satisfacer una necesidad** dada (EOQ).
- Grado en el que un conjunto de características inherentes **cumple con los requisitos** (ISO 9000:2000).
- => Es un **concepto relativo**:
  - La calidad está en los ojos del observador.
  - La calidad es relativa a las personas, su edad y circunstancias, al espacio, tiempo, etc.



## Introducción – Concepto de Calidad

- La **Calidad** es un concepto:

- **Multidimensional**
  - Referida a varias cualidades: Funcionalidad, Oportunidad, Coste, ...
- Sujeta a **restricciones**
  - Presupuesto disponible
- Ligado a **compromisos aceptables**
  - Plazos de fabricación
- No es ni totalmente subjetiva (porque ciertos aspectos pueden medirse) ni totalmente objetiva (ya que existen cualidades cuya evaluación sólo puede ser subjetiva).





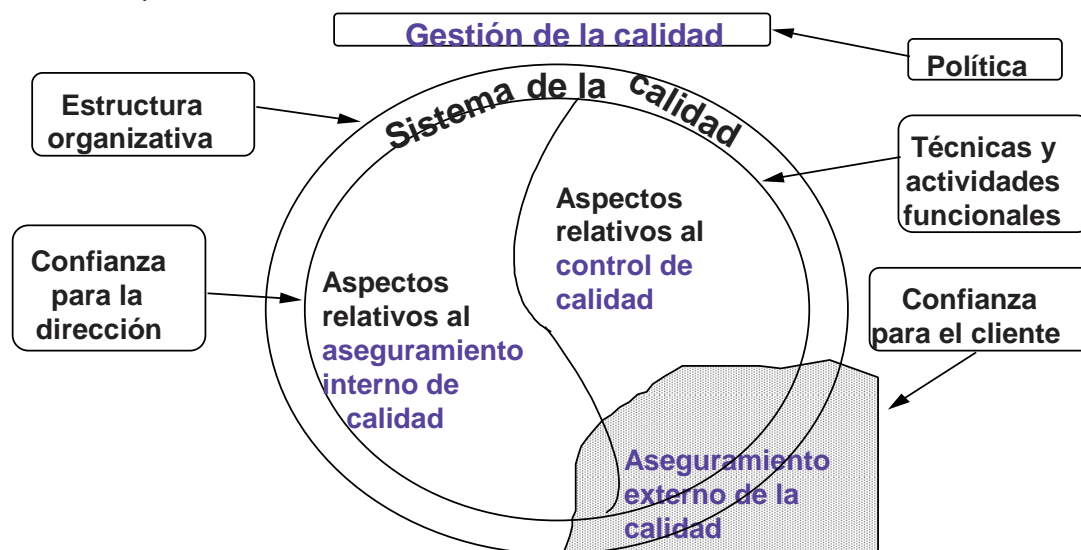
## Introducción – Concepto de Calidad

- 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**).



## Introducción – Concepto de Calidad

- Al hablar de Calidad a nivel de una Organización se manejan varios términos y conceptos (AENOR, 2000):





## Introducción – Concepto de Calidad

- Al hablar de Calidad a nivel de una Organización se manejan varios términos y conceptos (AENOR, 2000):
- **Gestión de la Calidad** (*Quality Management*):
  - Actividades coordinadas para **dirigir y controlar una organización en lo relativo a la calidad**.
  - Aspecto de la función general de la gestión que determina y aplica la política de calidad (objetivos y directrices generales de calidad de una empresa).
  - Normalmente se aplica a nivel de empresa => planificación estratégica, asignación de recursos, etc.
  - También puede haber una gestión de calidad dentro de la gestión de cada proyecto.



## Introducción – Concepto de Calidad

- Al hablar de Calidad a nivel de una Organización se manejan varios términos y conceptos (AENOR, 2000):
- **Aseguramiento de la Calidad** (*Quality Assurance*):
  - Parte de la gestión de la calidad orientada a **proporcionar confianza en que se cumplirán los requisitos de la calidad**.
  - Se debe evitar el término "garantía de calidad", ya que puede llevar a confusión con el concepto tradicional de garantía de los productos ("si falla, cubrimos los gastos o le devolvemos su dinero").
  - Pretende dar confianza en que el producto tiene calidad.



## Introducción – Concepto de Calidad

- Al hablar de Calidad a nivel de una Organización se manejan varios términos y conceptos (AENOR, 2000):
- **Control de la Calidad** (*Quality Control*):
  - Parte de la gestión de la calidad orientada al **cumplimiento de los requisitos de la calidad**.
  - Suele incluir 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 de un producto o servicio.
  - En general, son las actividades para evaluar la calidad de los productos desarrollados.



## Introducción – Perspectivas de la Calidad

- La Calidad se puede considerar desde varios **puntos de vista** diferentes:
  - **TRASCENDENTAL** (calidad = excelencia innata)
  - BASADA EN **USUARIO** (adecuación al propósito)
  - BASADA EN **FABRICANTE** (capacidad y madurez de proceso)
  - BASADA EN **PRODUCTO** (conformidad con requisitos)
  - BASADA EN **VALOR** (precio asequible)

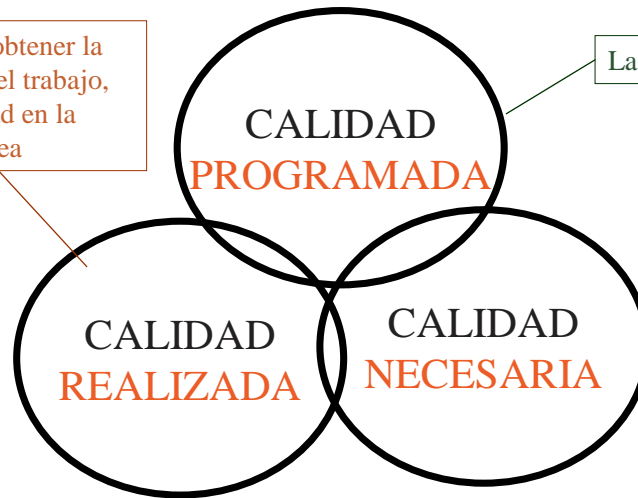


## Introducción – Perspectivas de la Calidad

- Y con diferentes **niveles**:

"Grado en el que un conjunto de características inherentes cumple con los requisitos" [ISO]

La que es capaz de obtener la persona que realiza el trabajo, gracias a su habilidad en la ejecución de una tarea



La que se ha pretendido obtener

La que el cliente exige con mayor o menor grado de concreción o, al menos, la que le gustaría recibir



## Introducción – Perspectivas de la Calidad

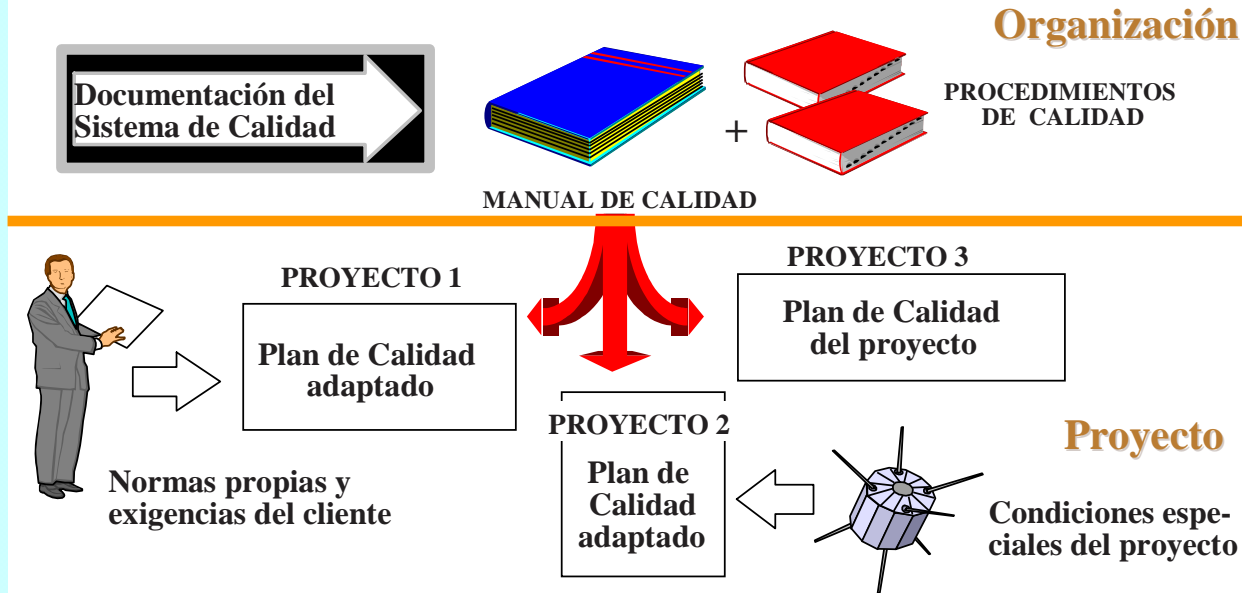
- La gestión de la calidad busca conseguir que estos tres círculos coincidan entre sí.
  - Todo lo que esté fuera de dicha coincidencia será motivo de derroche, de gasto superfluo o de insatisfacción.





## Introducción – Perspectivas de la Calidad

### LOS DOS NIVELES DE LA CALIDAD Y LAS RELACIONES ENTRE AMBOS



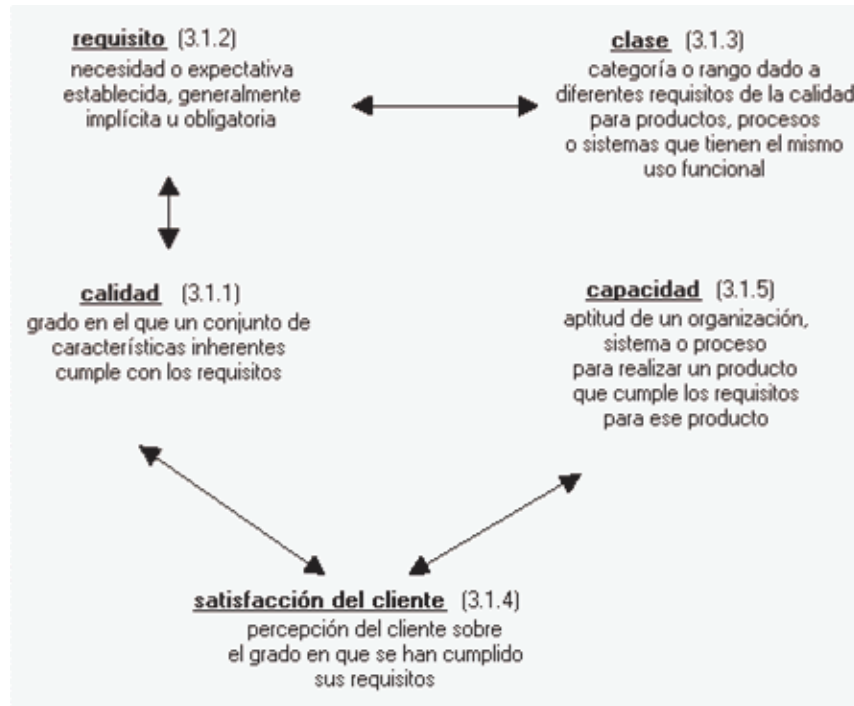
## Introducción – ISO 9000:2000

- La **norma ISO 9000:2000** es la principal referencial internacional sobre conceptos de calidad.
  - versión española:  
*UNE-EN ISO 9000:2000 Sistemas de Gestión de la Calidad. Fundamentos y Vocabulario*



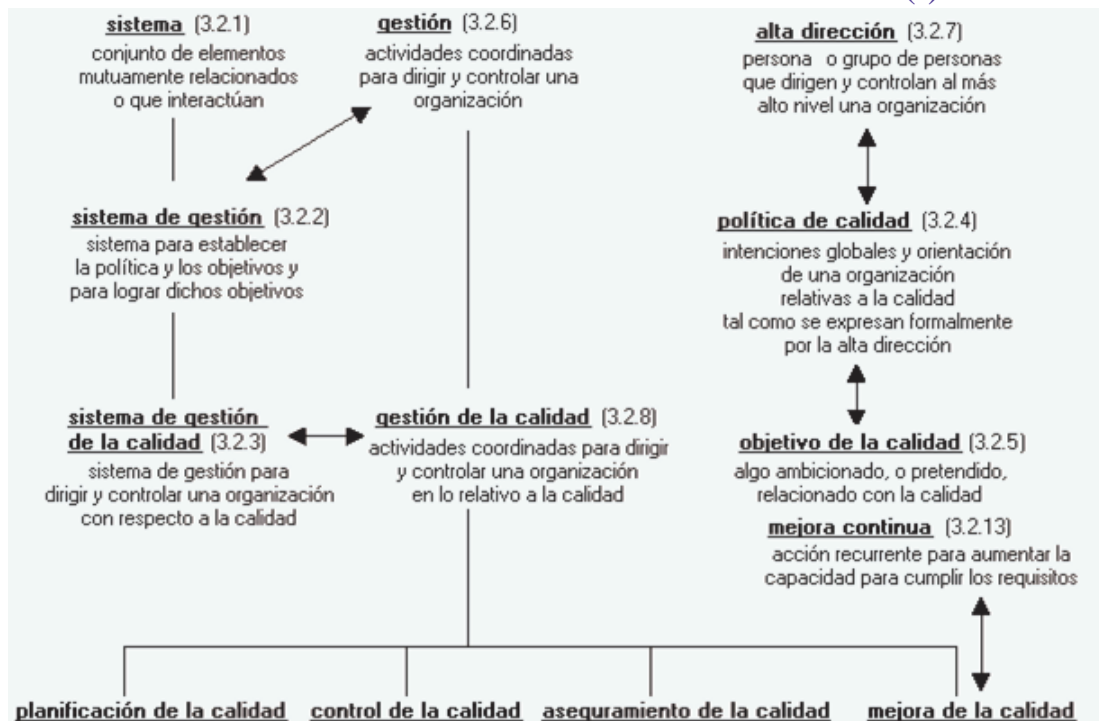
# Introducción – ISO 9000:2000

## TÉRMINOS RELATIVOS A LA CALIDAD



# Introducción – ISO 9000:2000

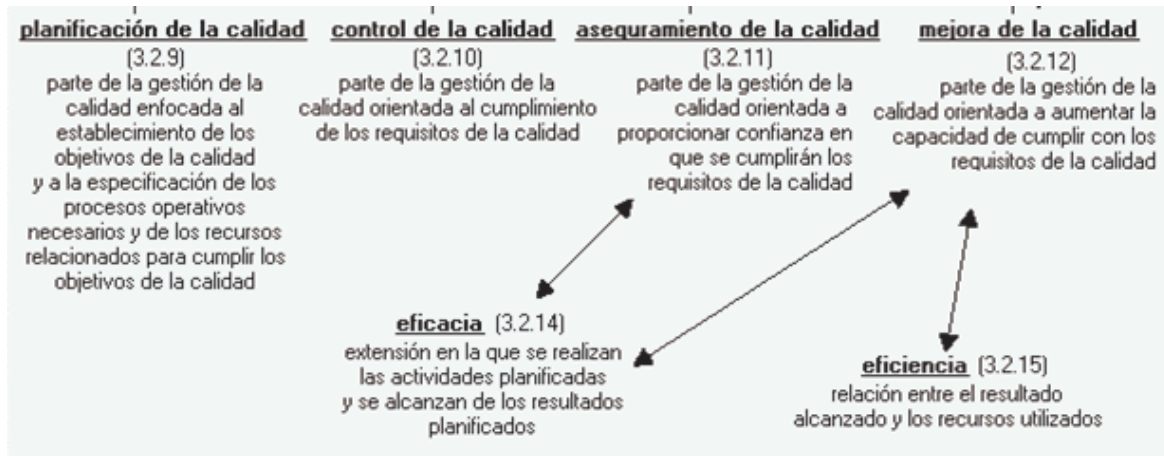
## TÉRMINOS RELATIVOS A LA GESTIÓN (i)





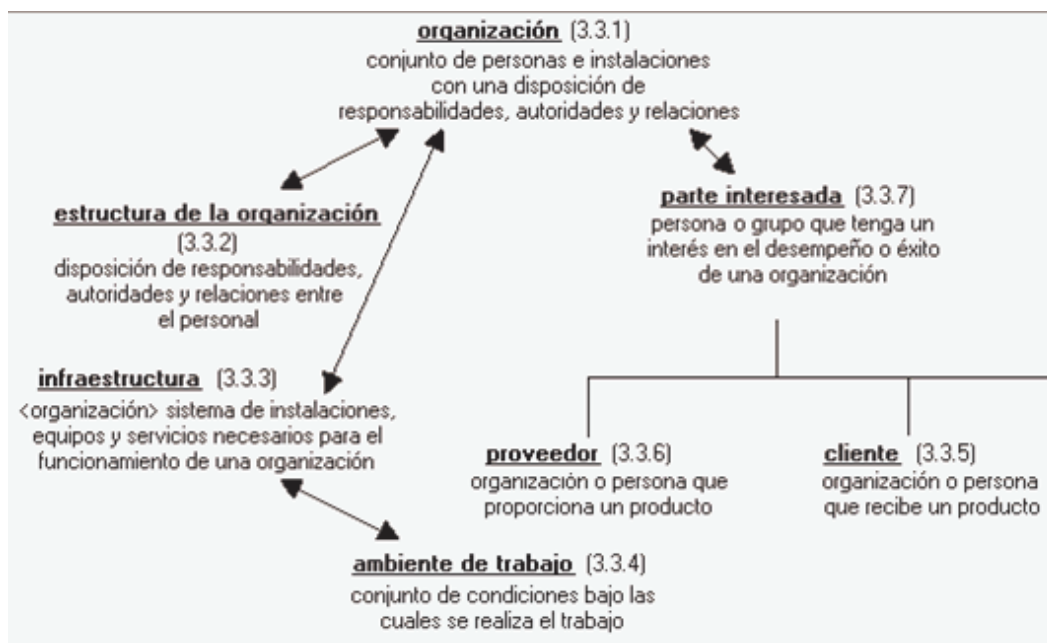
# Introducción – ISO 9000:2000

## TÉRMINOS RELATIVOS A LA GESTIÓN (ii)



# Introducción – ISO 9000:2000

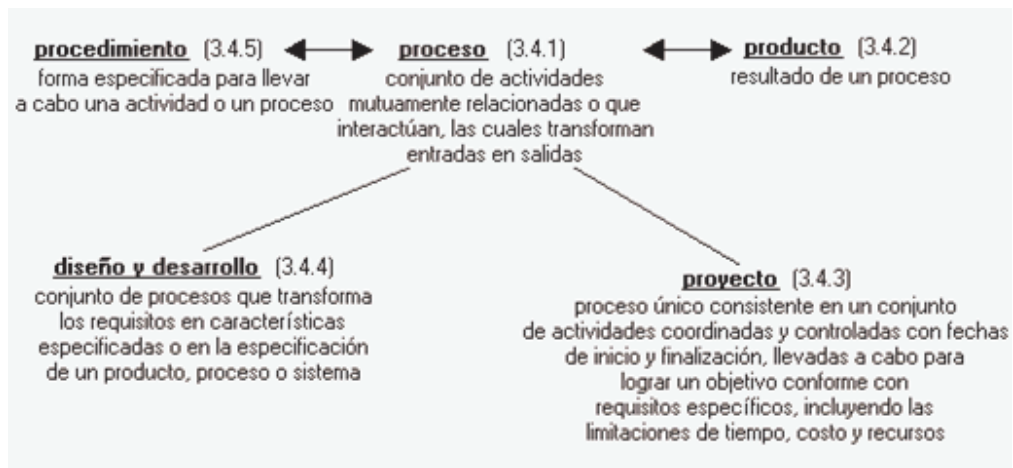
## TÉRMINOS RELATIVOS A LA ORGANIZACIÓN





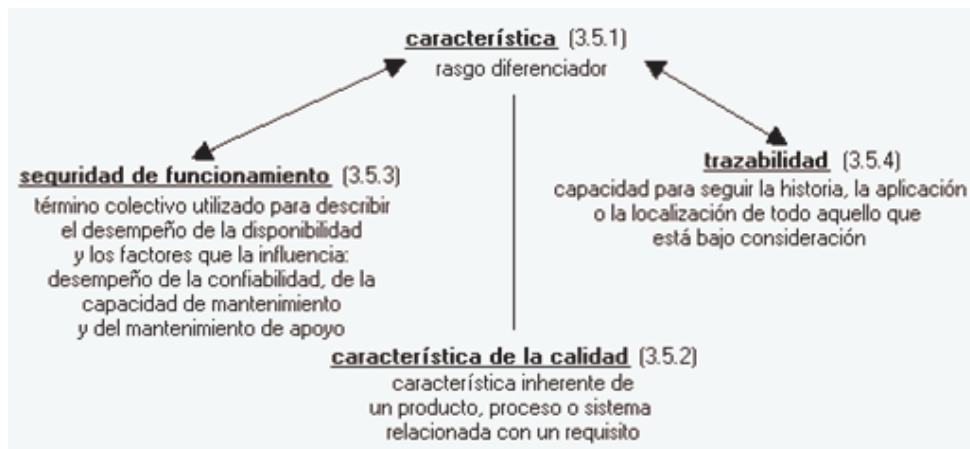
## Introducción – ISO 9000:2000

### TÉRMINOS RELATIVOS AL PROCESO Y AL PRODUCTO



## Introducción – ISO 9000:2000

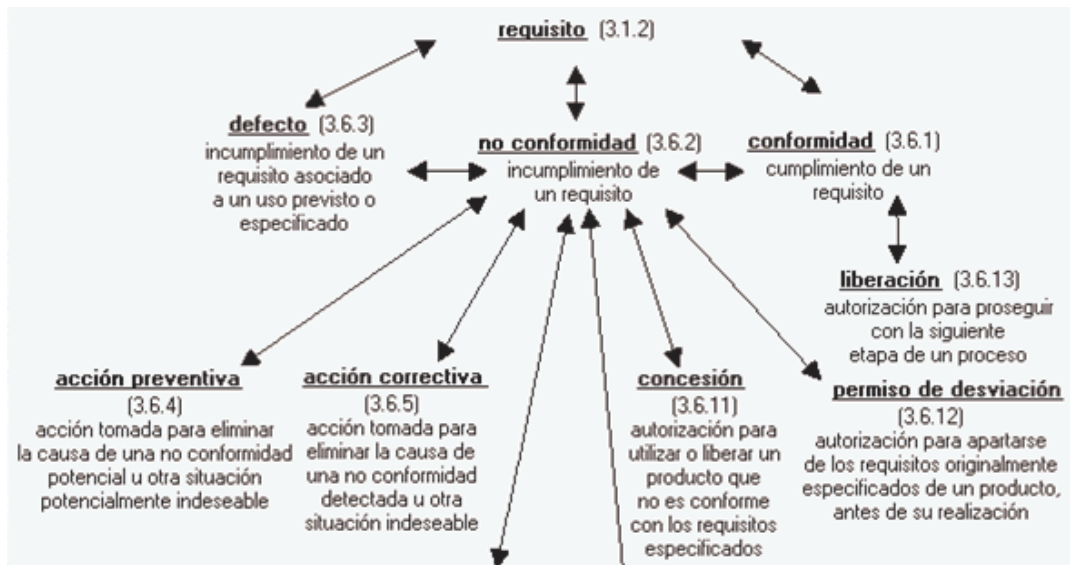
### TÉRMINOS RELATIVOS A LAS CARACTERÍSTICAS





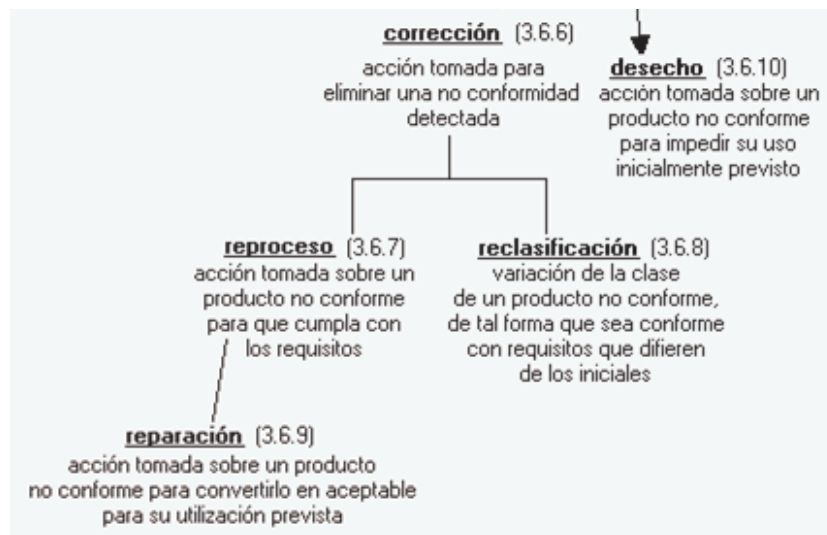
## Introducción – ISO 9000:2000

### TÉRMINOS RELATIVOS A LA CONFORMIDAD (i)



## Introducción – ISO 9000:2000

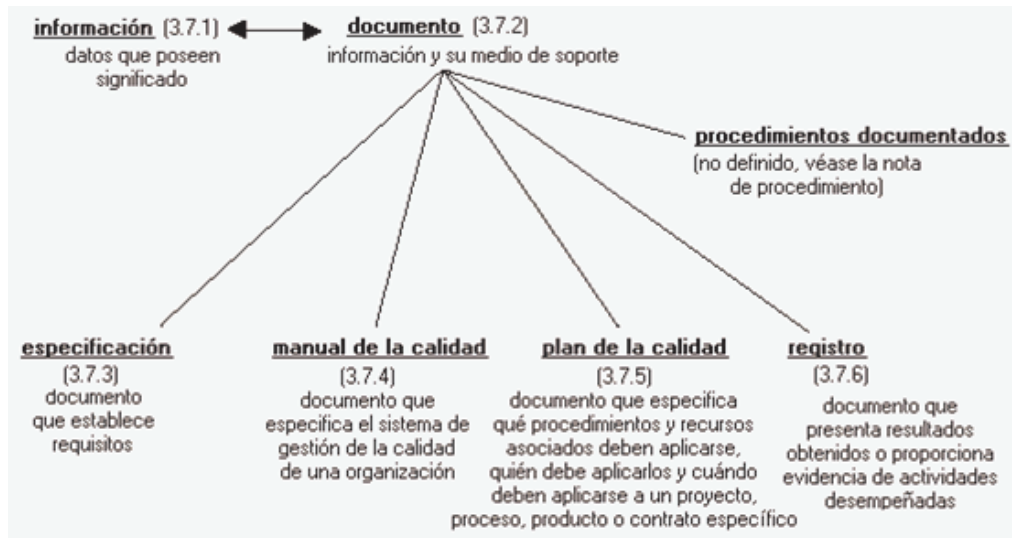
### TÉRMINOS RELATIVOS A LA CONFORMIDAD (ii)





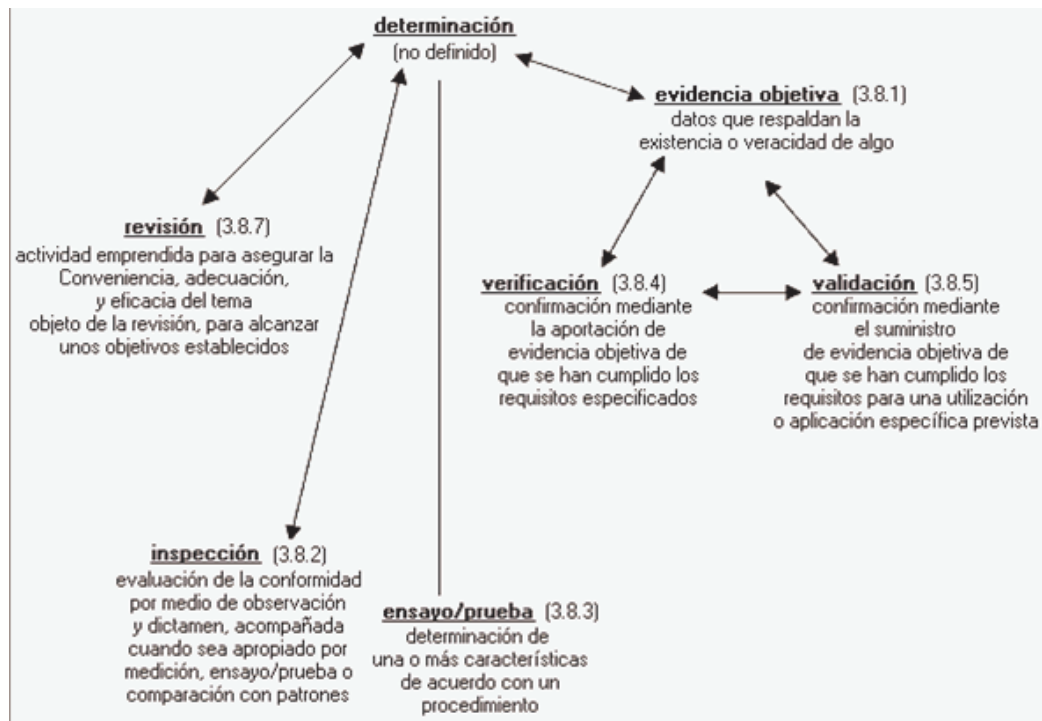
## Introducción – ISO 9000:2000

### TÉRMINOS RELATIVOS A LA DOCUMENTACIÓN



## Introducción – ISO 9000:2000

### TÉRMINOS RELATIVOS AL EXAMEN





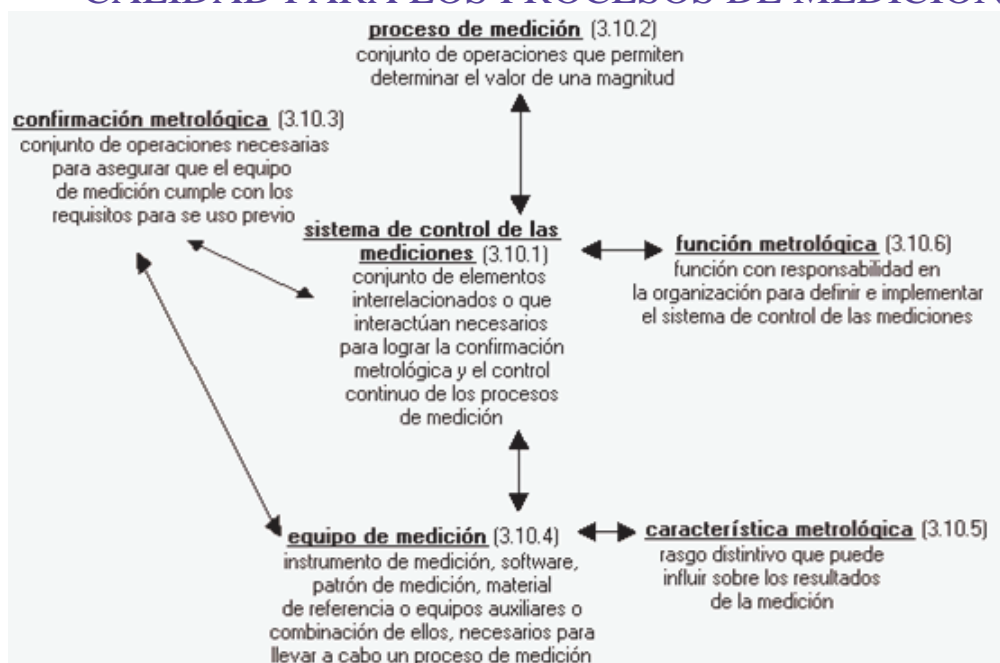
# Introducción – ISO 9000:2000

## TÉRMINOS RELATIVOS A LA AUDITORÍA



# Introducción – ISO 9000:2000

## TÉRMINOS RELATIVOS AL ASEGURAMIENTO DE LA CALIDAD PARA LOS PROCESOS DE MEDICIÓN



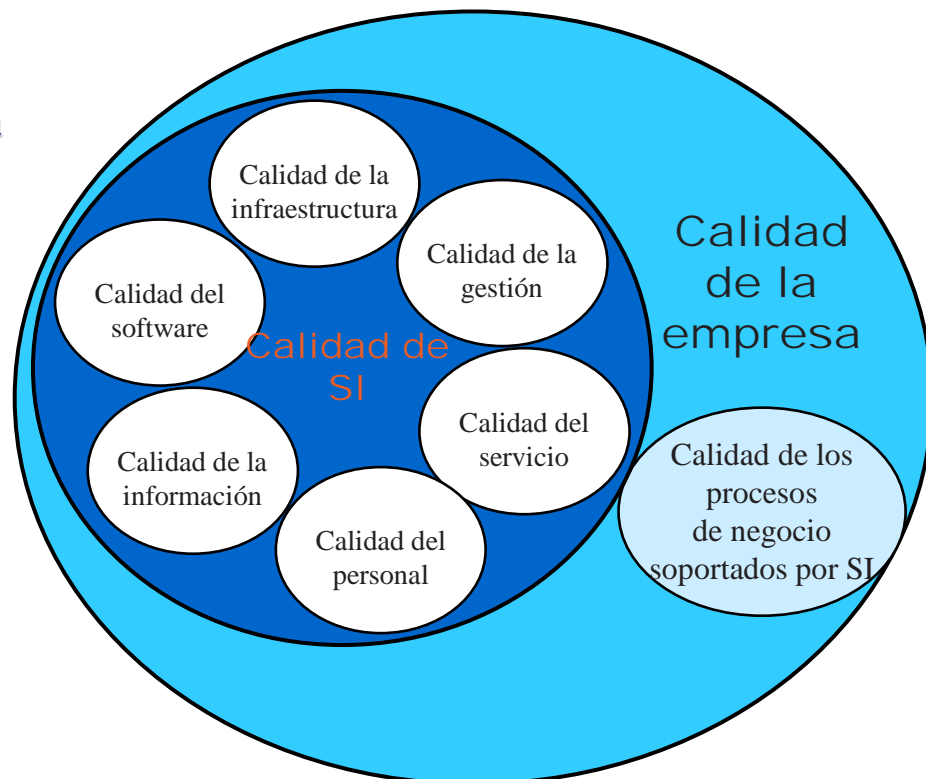


## Calidad de Sistemas de Información



## Calidad de Sistemas de Información

### Visión holística de la calidad

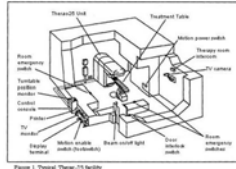




# Calidad del Software

## • Calidad en Ingeniería del Software:

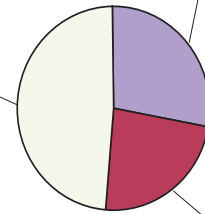
- Hombre con 103 años requerido a llevar a sus padres (2002)
- Mars Climate Orbiter (1999)
- Ariane 5 (1996)
- USS Yorktown (1998)
- Sistema de Manejo Automático de Equipaje - Denver (1994)
- Therac-25 (1985-1987)



Éxito en Proyectos Software (*informe CHAOS*)



Terminados con Deficiencias 49%



Terminados Con Éxito 28%

Terminados con Fracaso 23%

1.35



# Calidad del Software

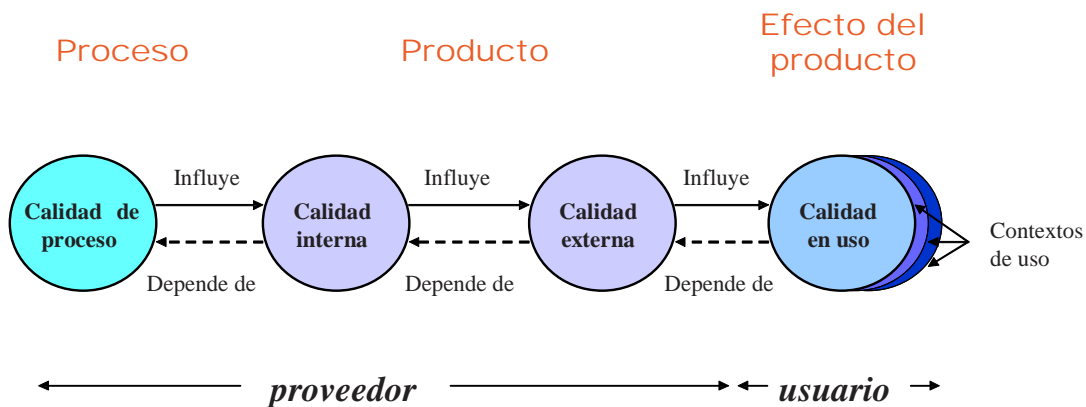
## • Calidad en Ingeniería del Software:

- El Software tiene características muy peculiares:
  - Se desarrolla, no se fabrica en el sentido clásico del término
  - Se trata de un producto lógico (ideas), sin existencia física
  - No se degrada con el uso
  - Suele ser complejo
  - Es muy flexible
- No olvidemos que el software no es sólo código
  - La calidad se obtiene a medida que se construye el producto
  - No hay que esperar a añadirla al final

Calidad es el grado con el que un sistema, componente o proceso cumple los requisitos especificados y/o las necesidades o expectativas del cliente o usuario [ IEEE Std.610-1991]



## Calidad del Software - Factores



## Calidad del Software - Factores

### • Diferentes aspectos de la calidad

- **Interna:** medible a partir de las características intrínsecas.
  - Código fuente.
- **Externa:** medible en el comportamiento del producto.
  - Resultados de una prueba.
- **En uso:** durante la utilización efectiva por parte del usuario.
  - Encuesta de satisfacción.

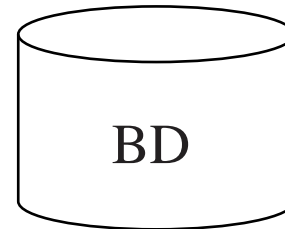
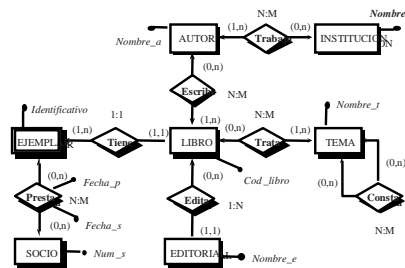


# Calidad del Software - Factores

*Calidad del proceso*



*Calidad del producto*

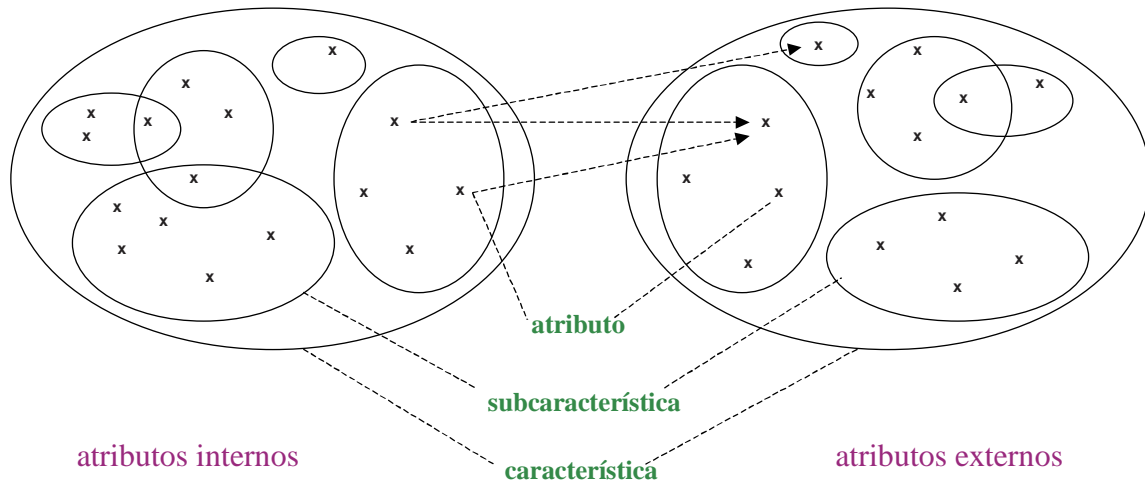


# Calidad de Producto – ISO 9126

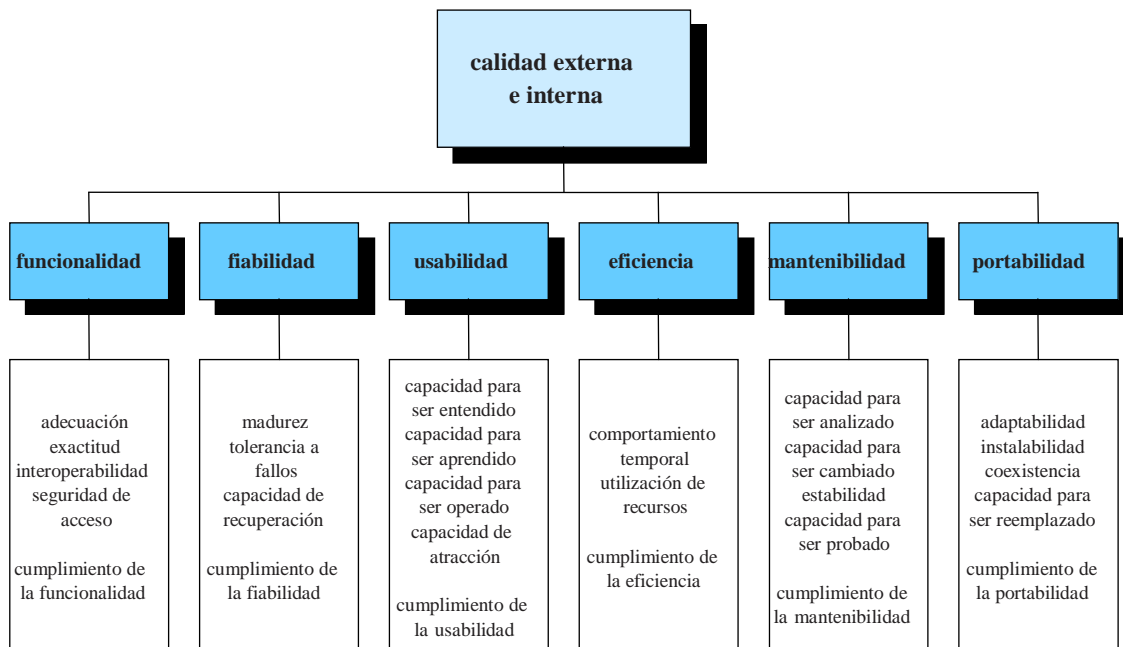
- **ISO/IEC 9126:** Tecnologías de la Información – Calidad de los Productos Software.
  - Parte 1: Modelo de Calidad
  - Parte 2: Métricas Externas
  - Parte 3: Métricas Internas
  - Parte 4: Métricas de Calidad en Uso
- **Utilidades:**
  - Validar la completitud de una definición de requisitos.
  - 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.



## Características, subcaracterísticas y atributos de calidad



## • Modelo de Calidad Interna y Externa





## Calidad de Producto – ISO 9126

- **Características del Modelo de Calidad:**
  - *Capacidad del producto software para ... bajo condiciones especificadas.*
  - **FUNCIONALIDAD:** Proporcionar funciones que satisfacen necesidades declaradas e implícitas cuando se usa.
  - **FIABILIDAD:** Mantener un nivel especificado de prestaciones cuando se usa.
  - **USABILIDAD:** Ser entendido, aprendido, usado y ser atractivo para el usuario, cuando se usa.
  - **EFICIENCIA:** Proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados.
  - **MANTENIBILIDAD:** Ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación a cambios en el entorno, requisitos o especificaciones funcionales.
  - **PORTABILIDAD:** Ser transferido de un entorno a otro.



## Calidad de Producto – ISO 9126

- **Subcaracterísticas de la Funcionalidad:**
  - *Capacidad del producto software para ... bajo condiciones especificadas.*
  - **Adecuación:** Proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.
  - **Exactitud:** Proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.
  - **Interoperabilidad:** Interactuar con uno o más sistemas especificados.
  - **Seguridad de acceso:** 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:** Adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.



## Calidad de Producto – ISO 9126

- Subcaracterísticas de la **Fiabilidad**:
  - *Capacidad del producto software para ... bajo condiciones especificadas.*
  - **Madurez**: Evitar fallar como resultado de fallos en el software.
  - **Tolerancia a fallos**: Mantener un nivel especificado de prestaciones en caso de fallos software o de infringir sus interfaces especificados.
  - **Capacidad de recuperación**: Reestablecer un nivel de prestaciones especificado y de recuperar los datos directamente afectados en caso de fallo.
  - **Cumplimiento de la fiabilidad**: Adherirse a normas, convenciones o regulaciones relacionadas con la fiabilidad.



## Calidad de Producto – ISO 9126

- Subcaracterísticas de la **Usabilidad**:
  - *Capacidad del producto software para ... bajo condiciones especificadas.*
  - **Entendibilidad**: Permitir al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.
  - **Aprendibilidad**: Permitir al usuario aprender sobre su aplicación.
  - **Operatividad**: Permitir al usuario operarlo y controlarlo.
  - **Atracción**: Ser atractivo al usuario.
  - **Cumplimiento de la usabilidad**: Adherirse a normas, convenciones, guías de estilo o regulaciones relacionadas con la usabilidad.



## Calidad de Producto – ISO 9126

- Subcaracterísticas de la **Eficiencia**:
  - *Capacidad del producto software para ... bajo condiciones especificadas.*
  - **Comportamiento temporal**: Proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados.
  - **Utilización de recursos**: Usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función.
  - **Cumplimiento de la eficiencia**: Adherirse a normas o convenciones relacionadas con la eficiencia.



## Calidad de Producto – ISO 9126

- Subcaracterísticas de la **Mantenibilidad**:
  - *Capacidad del producto software para ... bajo condiciones especificadas.*
  - **Analizabilidad**: Ser diagnosticadas deficiencias o causas de los fallos en el software, o para identificar las partes que han de ser modificadas.
  - **Cambiabilidad**: Permitir que una determinada modificación sea implementada.
  - **Estabilidad**: Evitar efectos inesperados debidos a modificaciones del software.
  - **Facilidad de prueba**: Permitir que el software modificado sea validado.
  - **Cumplimiento de la mantenibilidad**: Adherirse a normas o convenciones relacionadas con la mantenibilidad.



## Calidad de Producto – ISO 9126

- Subcaracterísticas de la **Portabilidad**:
  - *Capacidad del producto software para ... bajo condiciones especificadas.*
  - **Adaptabilidad**: Ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.
  - **Instalabilidad**: Ser instalado en un entorno especificado.
  - **Coexistencia**: Coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes.
  - **Reemplazabilidad**: Ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.
  - **Cumplimiento de la portabilidad**: Adherirse a normas o convenciones relacionadas con la portabilidad.



## Calidad de Producto - Datos

- **Situación**
  - Los datos son uno de los **activos más importantes de las organizaciones**, ya que son **clave en la toma de decisiones** estratégicas u operativas.
  - Por eso se recopilan datos para ser más competitivos.
  - Tal cantidad de datos puede llevar a una situación problemática:
    - Datos inútiles e innecesarios.
    - Redundancia incontrolada de datos.
    - Gran cantidad de datos históricos caducados.



## Calidad de Producto - Datos

### • Consecuencias

- Los datos se convierten en **fuentes de problemas**:
  - Datos no usados,
  - Barreras en la accesibilidad de los datos,
  - Dificultades en la utilización de los datos y de la información.
- Estos problemas afectan negativamente al **rendimiento de los procesos de negocio de una organización** a varios niveles:
  - **Técnico**:
    - Errores en la implementación de almacenes de datos.
  - **Organizacional**:
    - Pérdida de clientes al estar insatisfechos.
    - Pérdidas financieras debido a desperdicios de recursos en términos de tiempo y de dinero y a una baja o escasa productividad.
    - Trabajadores descontentos y desmotivados.
  - **Legal**:
    - Dependiendo de ciertas leyes, como la LOPD.



## Calidad de Producto - Datos

### • Calidad de Datos

- Características que deben tener los datos como materias primas para que, utilizando un proceso de producción adecuado, se pueda generar un producto de información.

### • Calidad de Información

- Aquellas características que debería tener un Producto de Información (PI) para que su utilización sea adecuada, es decir, cumpla con los requisitos de usuario

### • Dimensiones de Calidad de Datos

- Son criterios que permiten juzgar la calidad de los datos desde un determinado punto de vista. Se definen en la norma **ISO 25012** (similar a la 9126 para el Software).



## Calidad de Producto - Datos

- La **Calidad de los Datos depende de:**
  - Los **propios datos (extensión)**
    - Influyen en la efectividad de los procesos de negocio (dependencia de la semántica de los negocios).
  - El **esquema de los datos (intensión)**
    - Ejemplo: Tablas no normalizadas convenientemente.
    - Influye en el ciclo de vida de los datos.
    - Puede no dar el soporte para los aspectos de calidad requerida por el usuario.
  - **Procesos técnicos sobre los datos (SGBD):**
    - Pueden no implementar mecanismos que aseguren que no se producen errores en los datos, o que los datos satisfagan los requisitos de los usuarios.
    - Pueden depender de la calidad de los procesos o de la utilización de ciertos recursos de la organización.
    - Están normalmente implementados sobre el SGBD y dependen del soporte que de a esos procesos.



## Calidad de Producto - Datos

### • Ejemplos de falta de calidad

Id	Título	Director	Año	Nro_ Remakes	AñoUltimo Remake
1	Casablanca	Weir	1942	3	1940
2	El Club de los Poetas	Curtiz	1989	0	NULL
3	Vacaciones en Roma	Wylder	1953	0	NULL
4	Sabrina	NULL	1964	0	1985

No existe esta película, sino "El Club de los Poetas Muertos"

Curtiz es el director de Casablanca y Weir el de "El club de los Poetas Muertos"

Un remake no puede haberse hecho antes que la primera versión de la película

Falta el nombre del Director o no existe (hecho imposible o no se sabía)

Si el número de remakes es 0, no tiene sentido que haya una fecha para el último remake: o realmente se han hecho remakes o no debería aparecer una fecha



## Calidad de Proceso – ISO 90003

- La norma **ISO 90003** proporciona, a las organizaciones, una guía para la **adaptación de la ISO 9001:2000** para la adquisición, suministro, desarrollo, instalación y mantenimiento de **SOFTWARE** y servicios de soporte.
  - Identifica todos los aspectos que deberían ser tratados.
  - Es independiente de la tecnología, modelos de ciclo de vida, procesos de desarrollo y estructuras organizacionales.

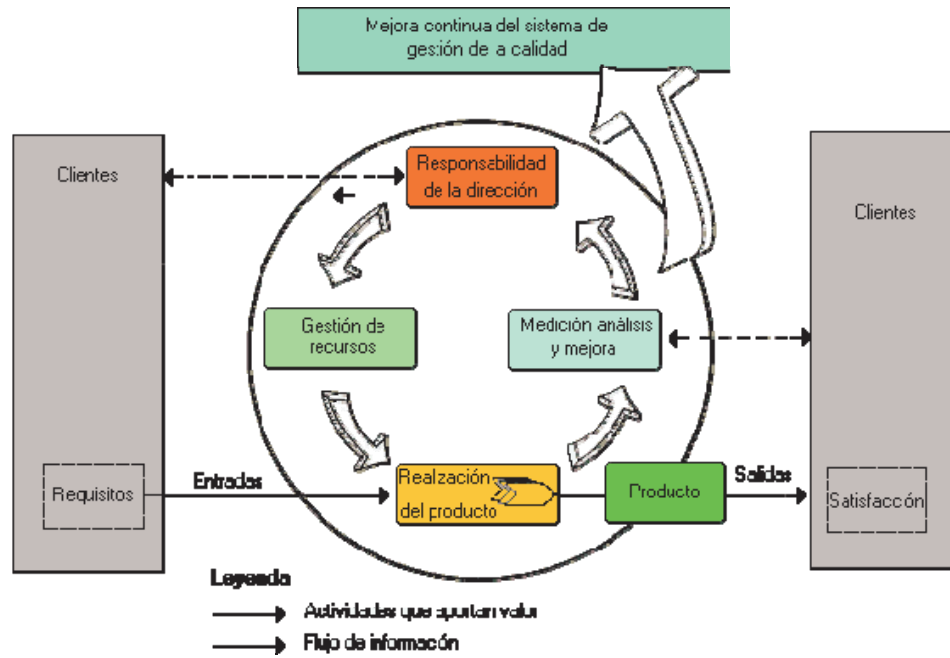


## Calidad de Proceso – ISO 90003

- La organización debe establecer, documentar, implementar y mantener un **sistema de gestión de la calidad software** y **mejorar continuamente** su eficacia, de acuerdo con los siguientes requisitos generales:
  - **Identificar los procesos** necesarios para el sistema de gestión de la calidad y su aplicación a través de la organización.
    - Identificar también los procesos de desarrollo, operación y mantenimiento de software.
  - Determinar la **secuencia e interacción** de estos procesos.
    - La organización debería también definir la secuencia e interacción de los procesos en los modelos de ciclos de vida del software, la planificación de la calidad y el desarrollo.
  - Determinar los **criterios y métodos** necesarios para asegurarse de que tanto la **operación** como el **control** de estos procesos sean **eficaces**.
  - Asegurarse de la **disponibilidad de recursos e información** necesarios para apoyar la operación y el seguimiento de estos procesos.
  - Realizar el **seguimiento, la medición y análisis** de estos procesos.
  - Implementar las acciones necesarias para alcanzar los resultados planificados y la **mejora continua** de estos procesos.

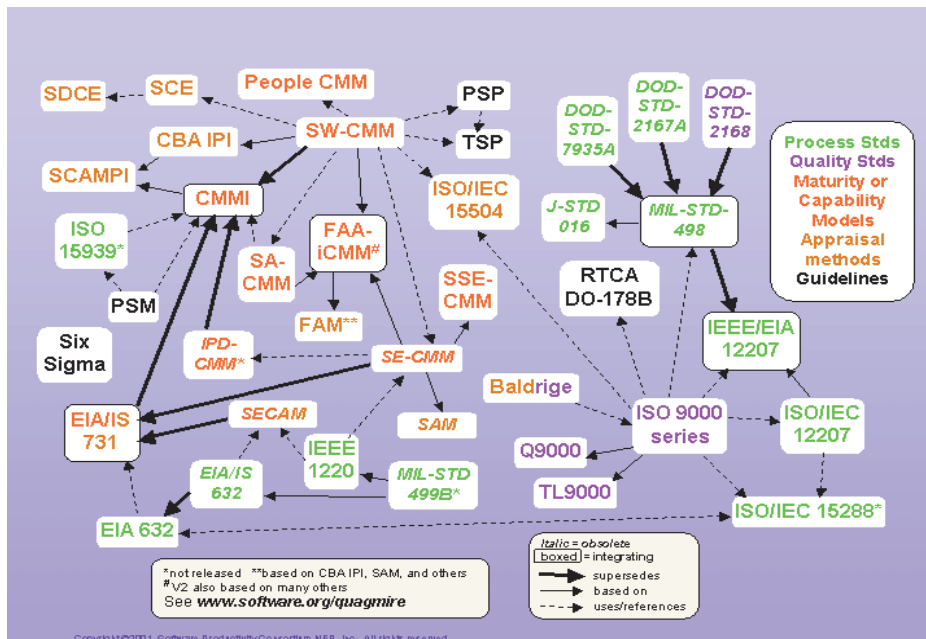


## Mejora continua del Sistema de Gestión de Calidad



## Evaluación y Mejora de Procesos

- Existen **multitud de normas** sobre procesos de ingeniería del software, su calidad y su mejora.





# Evaluación y Mejora de Procesos

- Existen **multitud de normas** ...

MODELO	URL
BOOTSTRAP (Kuvaja et al., 1994)	<a href="http://www.cse.dcu.ie/essiscope/sm5/approach/boot-2.html">http://www.cse.dcu.ie/essiscope/sm5/approach/boot-2.html</a>
EIA 632. Processes for Engineering a System, (Sheard y Lake, 1998)	<a href="http://www.eia.org">http://www.eia.org</a>
ISO/IEC 15504 (ISO, 2004 a-e)	<a href="http://www.iso.org">http://www.iso.org</a>
ISO/IEC 9003 (ISO/IEC, 2004f )	<a href="http://www.iso.org">http://www.iso.org</a>
MIL STD -498	<a href="http://www.pogner.demon.co.uk/mil_498/">http://www.pogner.demon.co.uk/mil_498/</a>
MOPROSOFT (Oktaba et al., 2003)	<a href="http://www.lania.mx/biblioteca/manuales/moprosoft/V%201.1%20DocumentoBase.pdf">www.lania.mx/biblioteca/manuales/moprosoft/V%201.1%20DocumentoBase.pdf</a>
Mps BPR (Weber y Rocha , 2004)	<a href="http://www.softex.br/">http://www.softex.br/</a>
SEI CMMI - Capability Maturity Model Integration (SEI, 2002)	<a href="http://www.sei.cmu.edu/cmmi/">http://www.sei.cmu.edu/cmmi/</a>
SCAMPI (Standard CMMI Appraisal Method for Process Improvement) (SEI, 2001)	<a href="http://www.sei.cmu.edu/publications/document_s/01.reports/01hb001.html">http://www.sei.cmu.edu/publications/document_s/01.reports/01hb001.html</a>
SEI Software Capability Evaluation (SCE) (Byrnes y Philips, 1996)	<a href="http://www.sei.cmu.edu/publications/document_s/96.reports/96.tr.002.html">http://www.sei.cmu.edu/publications/document_s/96.reports/96.tr.002.html</a>
SEI SE -CMM Capability Maturity Model for Systems Engineering (SEI, 1995)	<a href="http://www.sei.cmu.edu/cmm/se_-cmm.html">http://www.sei.cmu.edu/cmm/se_-cmm.html</a>
SEI P -CMM People Capability Maturity Model (Curtis et al., 2001)	<a href="http://www.sei.cmu.edu/cmm_-p/">http://www.sei.cmu.edu/cmm_-p/</a>
SEI IDEAL Model (Gremba y Myers, 1997)	<a href="http://www.sei.cmu.edu/ideal/ideal.html">http://www.sei.cmu.edu/ideal/ideal.html</a>
SEI Personal Software Process (PSP) (Humphrey, 1997)	<a href="http://www.sei.cmu.edu/tsp/psp.html">http://www.sei.cmu.edu/tsp/psp.html</a>
Systems Security Engineering Capability Maturity Model (SSE -CMM) ( Department of Defense U.S.A., 1999)	<a href="http://www.sse_-cmm.org">http://www.sse_-cmm.org</a>
SEI SW -CMM Capability Maturity Model SM for Software (SEI, 1995)	<a href="http://www.sei.cmu.edu/cmm/cmm.html">http://www.sei.cmu.edu/cmm/cmm.html</a>
SEI Team Software Process (TSP) (Humphrey, 2000 a; 2000b )	<a href="http://www.sei.cmu.edu/tsp/tsp.html">http://www.sei.cmu.edu/tsp/tsp.html</a>
Software Development Capability Evaluation (SDCE) (AFMC, 1994)	<a href="http://www.stsc.hill.af.mil/crosstalk/1997/04/development.asp">http://www.stsc.hill.af.mil/crosstalk/1997/04/development.asp</a>
Tickit (Tickit Project Office, 1992)	<a href="http://www.tickit.org/">http://www.tickit.org/</a>
Trillium (Trillium Team, 1994) (April y Coallier, 1995)	<a href="http://www2.umassd.edu/swpi/BellCanada/trillium-html/trillium.html">http://www2.umassd.edu/swpi/BellCanada/trillium-html/trillium.html</a>

Francisco Ruiz - IS2

1.59



# Evaluación y Mejora de Procesos - CMMI

- Capability Maturity Model Integration (CMMI)** es un **modelo para la mejora de procesos** que proporciona a las organizaciones los elementos esenciales para procesos eficaces.
  - Las prácticas CMMI deben adaptarse a cada organización en función de sus objetivos de negocio.
- Sirve para **evaluar**:
  - Una organización es evaluada (por ejemplo, usando el método SCAMPI) y recibe una calificación de **nivel de madurez**, como **organización**, de 1-5.
  - Opcionalmente, puede ser evaluada para ciertos procesos en base a las llamadas **áreas de proceso**, para obtener el **perfil de capacidad** de la organización en cierto proceso.

Francisco Ruiz - IS2

1.60



## Evaluación y Mejora de Procesos - CMMI

- CMMI comprende **tres modelos**, incluido CMMI-SW:

Disciplina	Modelo Fuente	Descripción
<b>Software</b>	El CMMI para software (CMMI-SW)	El CMMI-SW (antiguo CMM) está organizado para ayudar a la organizaciones de software a mejorar mediante una trayectoria evolutiva, creciendo con fines específicos, desde un ambiente caótico hacia unos maduros y disciplinados procesos de software
<b>Ingeniería de Sistemas</b>	Modelo de Capacidad de Ingeniería de Sistemas (EIA/IS-731)	Integración de todas las disciplinas de sistemas para que conozcan las necesidades técnicas y de negocio de la forma más efectiva
<b>Proceso integrado de desarrollo de productos</b>	Desarrollo integrado de producto (CMMI-IPD)	Enfoque sistemático para el desarrollo del producto que incrementa la satisfacción del cliente mediante una colaboración oportuna de las disciplinas necesarias a lo largo del ciclo de vida del producto.



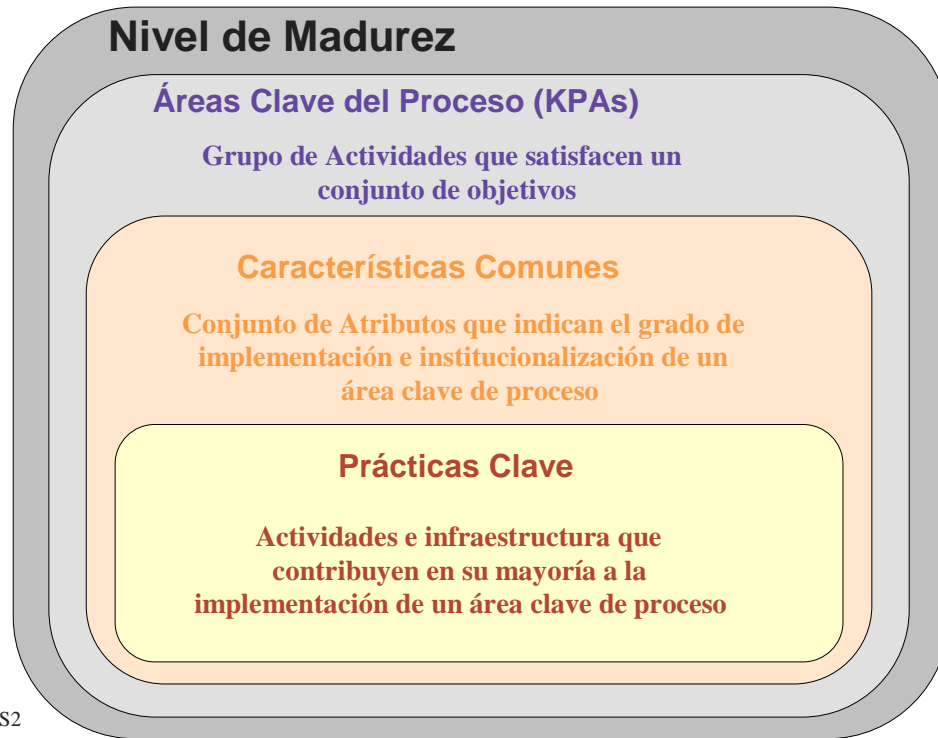
## Evaluación y Mejora de Procesos - CMMI

- CMMI es la evolución y ampliación de **CMM (Capability Maturity Model)**:
  - Ahora conocido **CMMI-SW** (de **software**)
  - Desarrollado por el SEI (*Software Engineering Institute*) de la Universidad de Carnegie Mellon, USA.
  - Proporciona a las organizaciones de software el **modelo de referencia** necesario como soporte para el **control** de sus **procesos de desarrollo y mantenimiento** y para facilitar su evolución hacia una cultura de la Ingeniería del Software y de excelencia en la gestión.
  - Sirve para dos cosas principales:
    - **Evaluar la madurez** de los procesos de desarrollo de software dentro de una organización.
    - **Proponer un plan de mejora** de los procesos de desarrollo de software de acuerdo a una serie de niveles.



## Evaluación y Mejora de Procesos - CMMI

- **Conceptos Básicos de CMMI-SW**



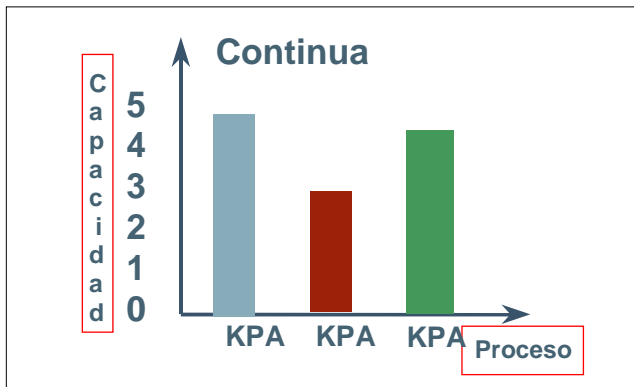
## Evaluación y Mejora de Procesos - CMMI

- Hay dos **representaciones** de **CMMI-SW**, es decir, dos maneras diferentes de utilizarlo.
  - Cada organización debe optar por una de las dos.
- **Escalonada** *[por etapas]*
  - Establece 5 **Niveles de Madurez** (Maturity Level) para clasificar a las organizaciones, en función de qué áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería.
  - Centrada en la **madurez de la organización**.
  - La selección de las áreas de proceso clave (KPA) está prefijada, habiendo 7 KPA para el nivel de madurez 2 (ML2), 11 para el ML3, 2 para el ML4 y 2 más para el ML5.
- **Continua**
  - Muestra la representación del nivel de **capacidad de la organización para cada una de las áreas de proceso**.



# Evaluación y Mejora de Procesos - CMMI

## • Representación Continua vs Escalonada



## EVALUACION DE PROCESOS

## CMMI

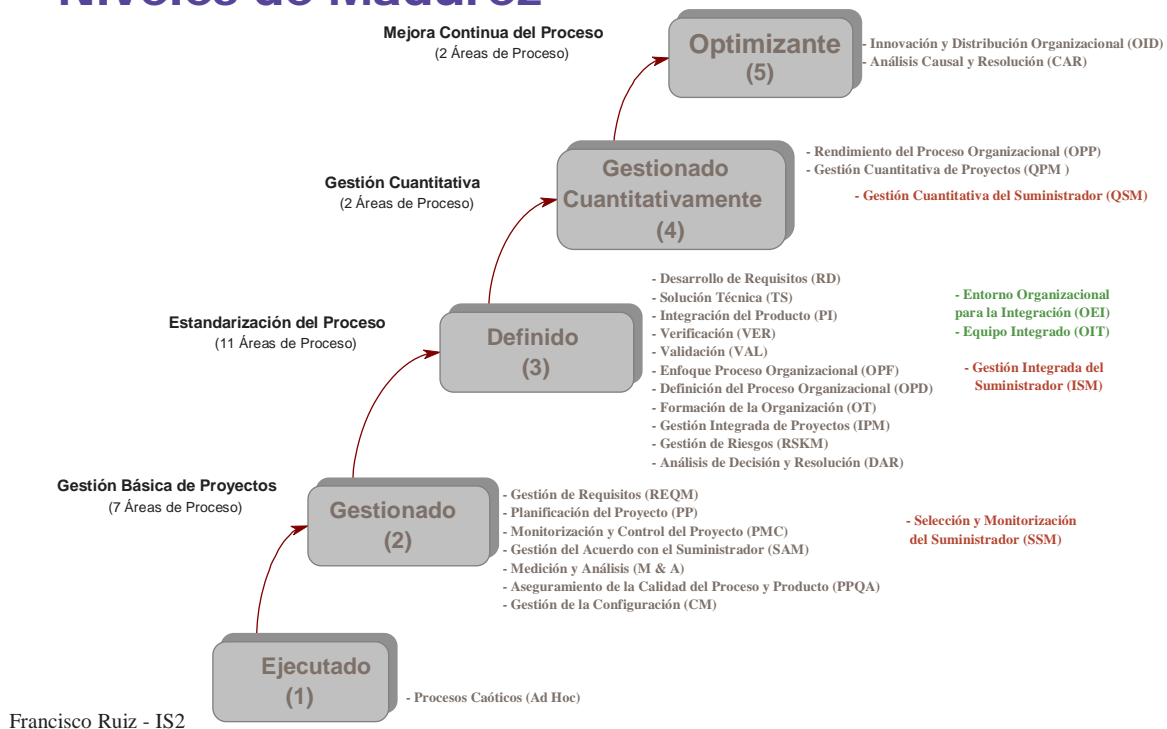
## • Áreas de Proceso Claves (KPAs)





# Evaluación y Mejora de Procesos - CMMI

## • Niveles de Madurez



Francisco Ruiz - IS2

1.67



# Evaluación y Mejora de Procesos - CMMI

## • Evaluación [appraisal]

- Las organizaciones pueden querer evaluar su nivel de madurez (organizacional) o su nivel de capacidad (de procesos determinados) por varias razones:
  - Para comparar con las mejores prácticas CMMI y determinar qué mejoras se pueden hacer.
  - Para informar a los clientes externos y proveedores acerca de cómo funciona la organización y lleva a cabo sus procesos (buena imagen y confianza).
  - Para cumplir los requisitos contractuales de uno o más clientes.

Francisco Ruiz - IS2

1.68



## Evaluación y Mejora de Procesos - CMMI

- **Evaluación** *[appraisal]*
  - **SCAMPI (Standard CMMI Appraisal Method for Process Improvement)** es el método propuesto por el SEI para realizar evaluaciones CMMI, con el fin de:
    - Identificar fortalezas y debilidades de los procesos,
    - Revelar riesgos de desarrollo/adquisición, y
    - Determinar niveles de capacidad y madurez.

<http://www.sei.cmu.edu/cmmi/appraisals/>

  - Se utiliza como parte de un proceso o programa de mejora, o para la calificación de posibles proveedores.
  - Define un proceso de evaluación con varias **fases**:
    - Preparación
    - Actividades sobre el terreno
    - Observaciones preliminares, conclusiones y valoraciones
    - Presentación de informes y actividades de seguimiento



## Evaluación y Mejora de Procesos - CMMI

- **Mejora** *[improvement]*
  - **IDEAL** define un marco de ciclo de vida para la mejora de procesos.

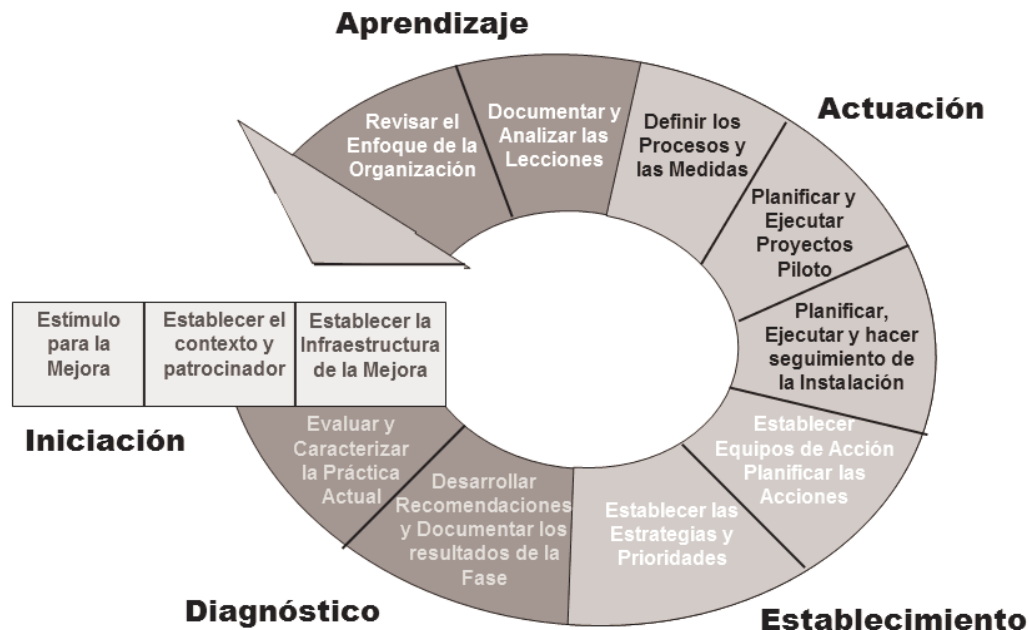
<http://www.sei.cmu.edu/ideal/>

  - Establece un modelo con cinco fases (iniciales en inglés):
    - Iniciación *(initiating)*
    - Diagnóstico *(diagnosing)*
    - Establecimiento *(establishing)*
    - Actuación *(acting)*
    - Aprendizaje *(learning)*



## Evaluación y Mejora de Procesos - CMMI

- **Mejora - Modelo IDEAL:**



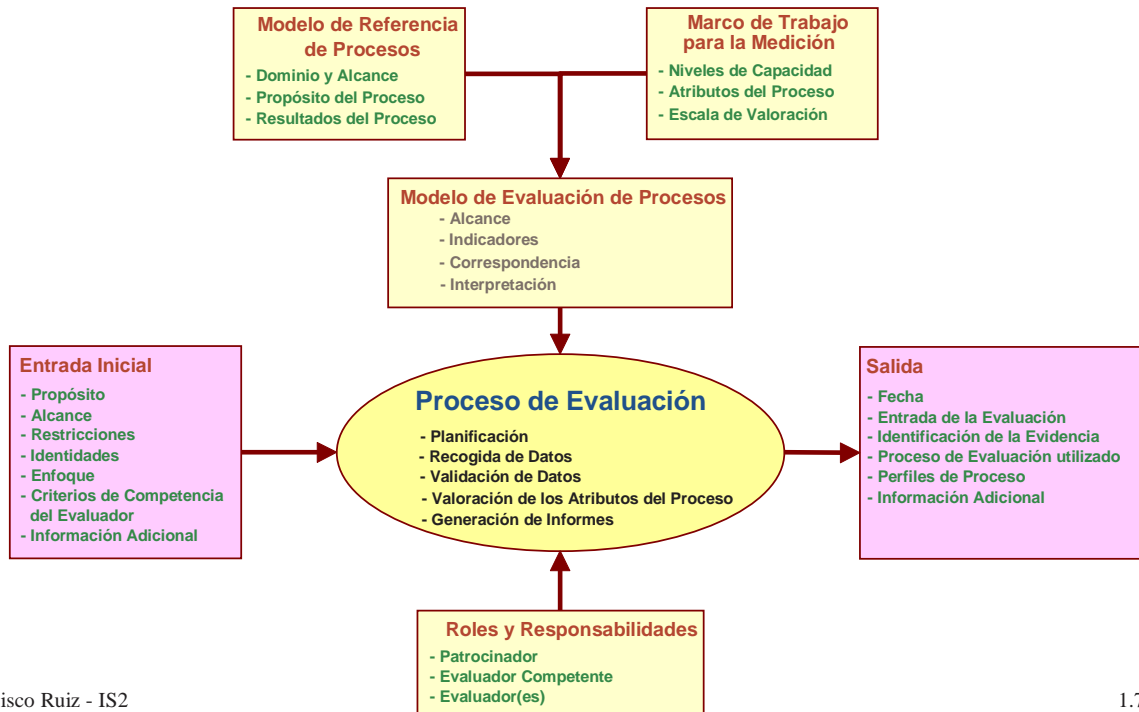
## Evaluación y Mejora de Procesos – ISO 15504

- El estándar **ISO/IEC 15504** proporciona:
  - Proporciona un **marco de trabajo para la evaluación de procesos software**, y
  - Establece los **requisitos mínimos para realizar una evaluación** que asegure la repetibilidad y consistencia de las valoraciones obtenidas
  - El objetivo de la evaluación del proceso es conocer la capacidad de los procesos de una organización.
- Frente a CMMI:
  - Ventajas:
    - Es estándar internacional oficial (alineado con los demás estándares ISO).
    - Es más completo y versátil.
  - Desventajas:
    - Está menos implantado a nivel industrial (lleva menos años).



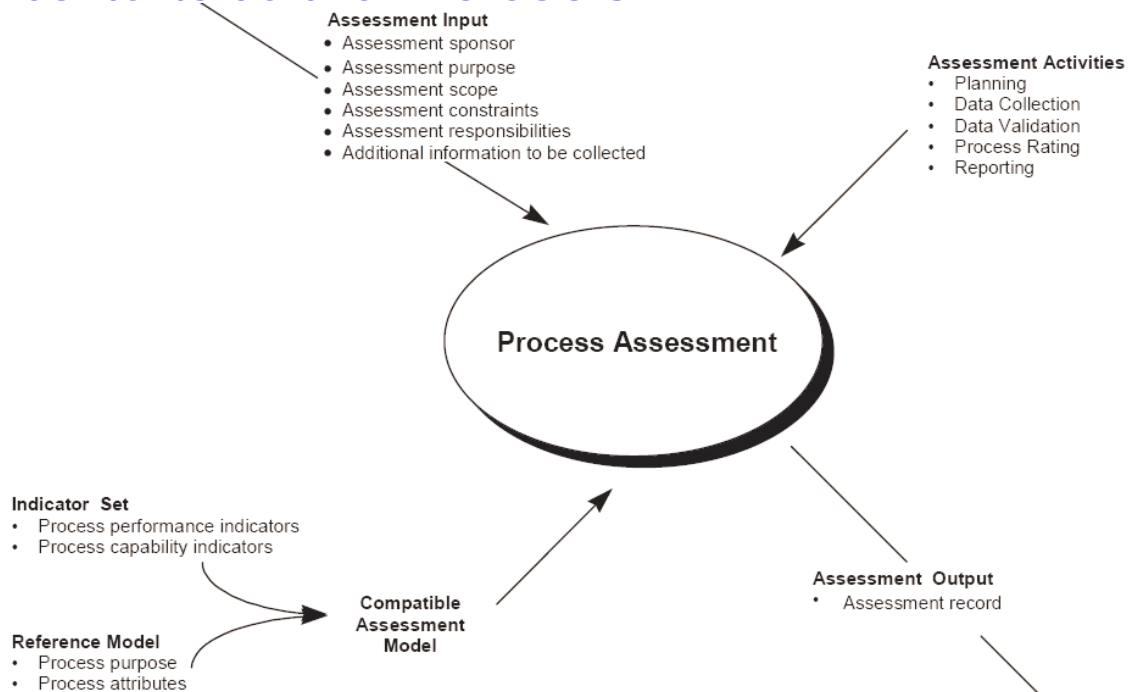
# Evaluación y Mejora de Procesos – ISO 15504

## Partes principales



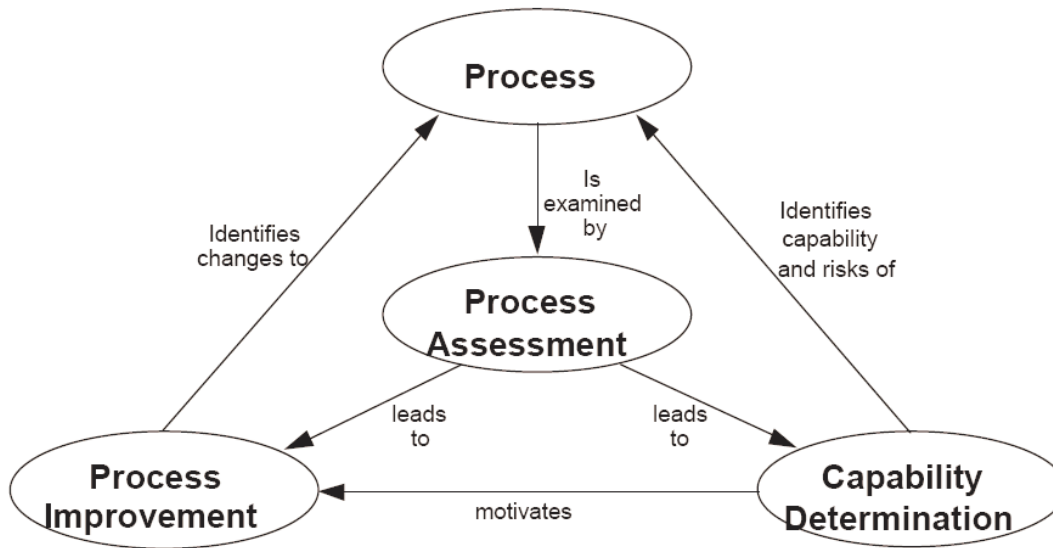
# Evaluación y Mejora de Procesos – ISO 15504

## Contexto de una Evaluación

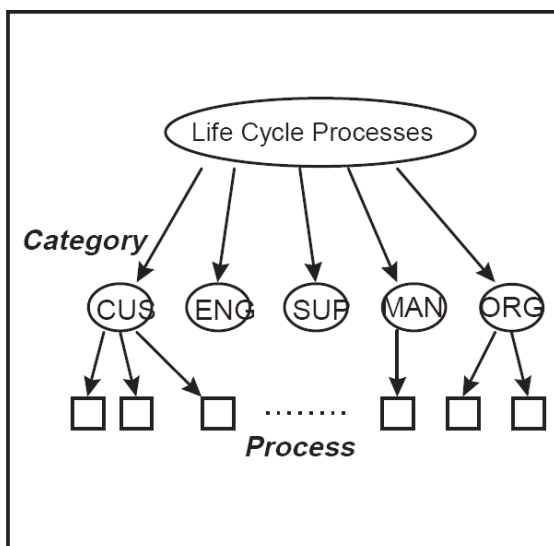




## • Mejora vs Capacidad



## • Dimensiones: Proceso vs Capacidad



**Process Dimension**

×  
are mapped against

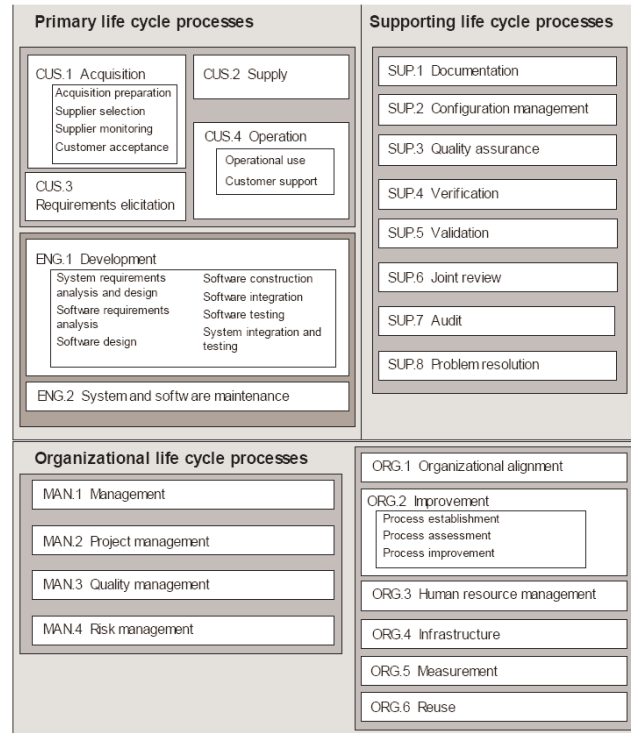
Level	Name	Attributes
5	<i>Optimizing process</i>	Process change attribute Continuous improvement attribute
4	<i>Predictable Process</i>	Process measurement attribute Process control attribute
3	<i>Established Process</i>	Process definition attribute Process resource attribute
2	<i>Managed Process</i>	Performance management attribute Work Product management attribute
1	<i>Performed Process</i>	Process performance attribute
0	<i>Incomplete Process</i>	

**Capability Dimension**



# Evaluación y Mejora de Procesos – ISO 15504

- Modelo de Referencia - **Procesos**
- según ISO 12207



# Evaluación y Mejora de Procesos – ISO 15504

- Modelo de Referencia - **Capacidad**
- Niveles de Capacidad
  - Atributos de Proceso



Process Attribute ID	Capability Levels and Process Attributes
	<b>Level 0 : Incomplete process</b>
	<b>Level 1 : Performed process</b>
PA 1.1	Process performance
	<b>Level 2 : Managed process</b>
PA 2.1	Performance management
PA 2.2	Work product management
	<b>Level 3 : Established process</b>
PA 3.1	Process definition
PA 3.2	Process resource
	<b>Level 4 : Predictable process</b>
PA 4.1	Measurement
PA 4.2	Process control
	<b>Level 5 : Optimizing process</b>
PA 5.1	Process change
PA 5.2	Continuous improvement



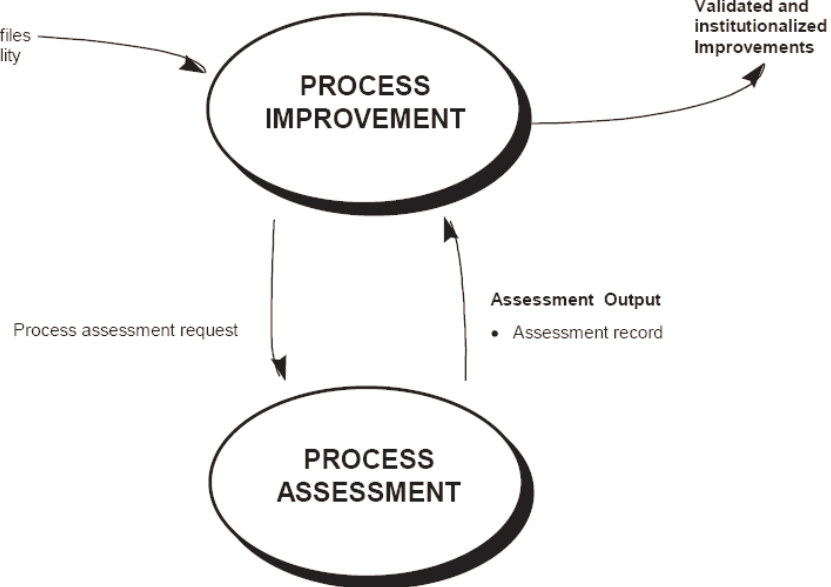
## • Contexto de una Mejora

Software process improvement request

Organization's needs and business goals

Target capability profiles from process capability determination

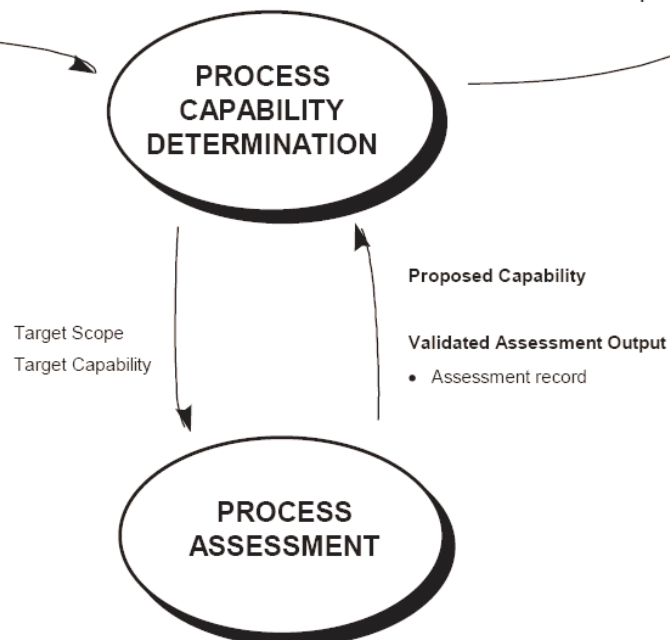
Industry norms and benchmarks



## • Contexto para Determinar la Capacidad

Specified Requirements

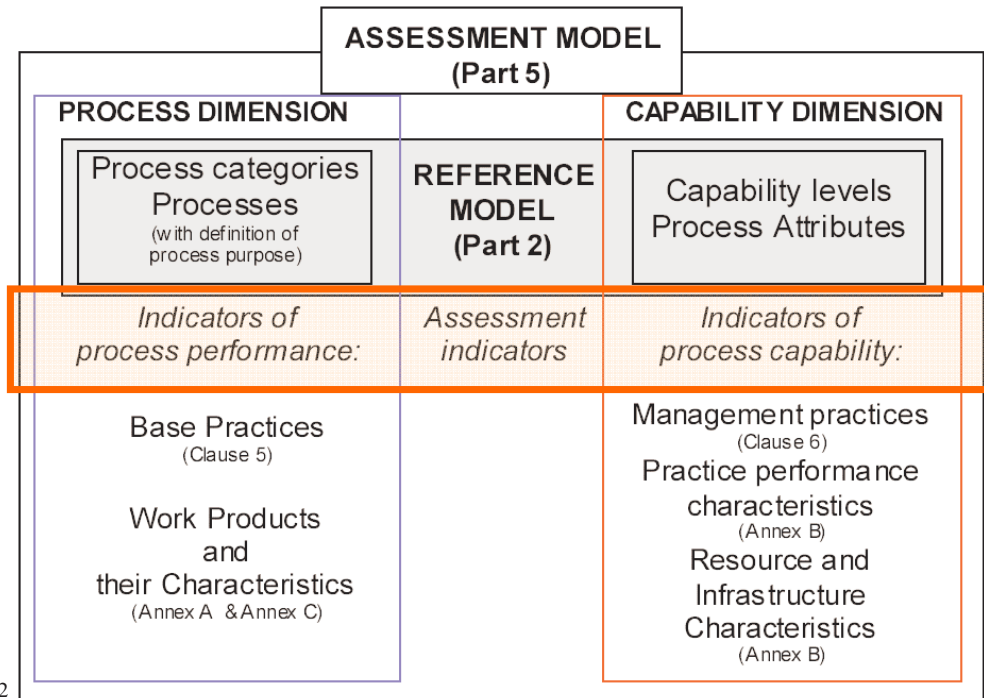
Capability Report





# Evaluación y Mejora de Procesos – ISO 15504

## • Modelo de Evaluación



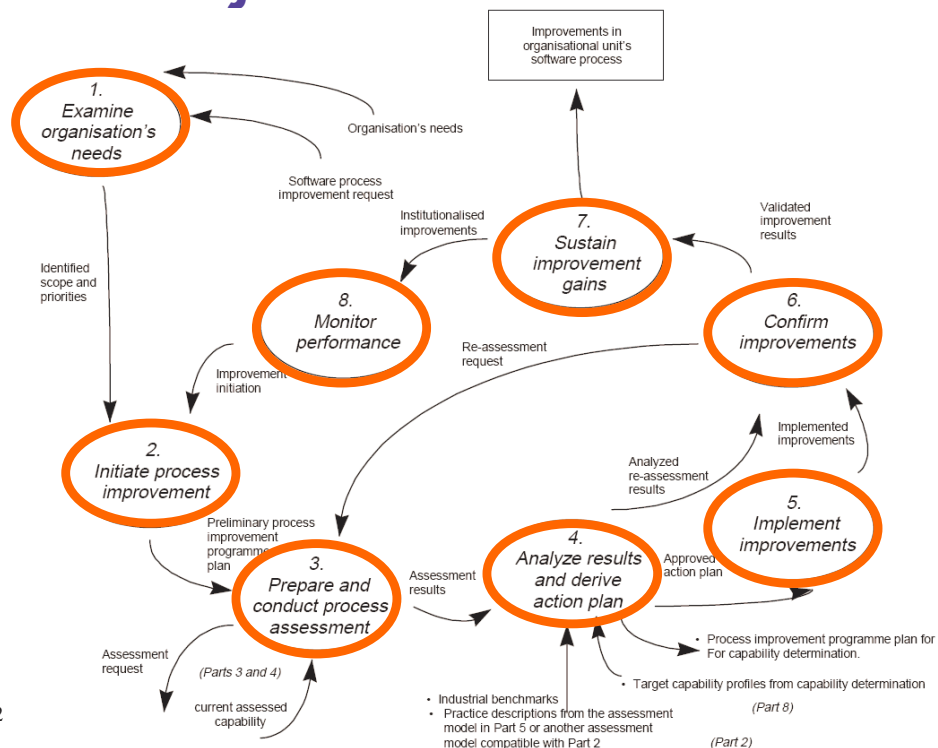
Francisco Ruiz - IS2

1.81



# Evaluación y Mejora de Procesos – ISO 15504

## • Etapas en la Mejora Continua de Procesos



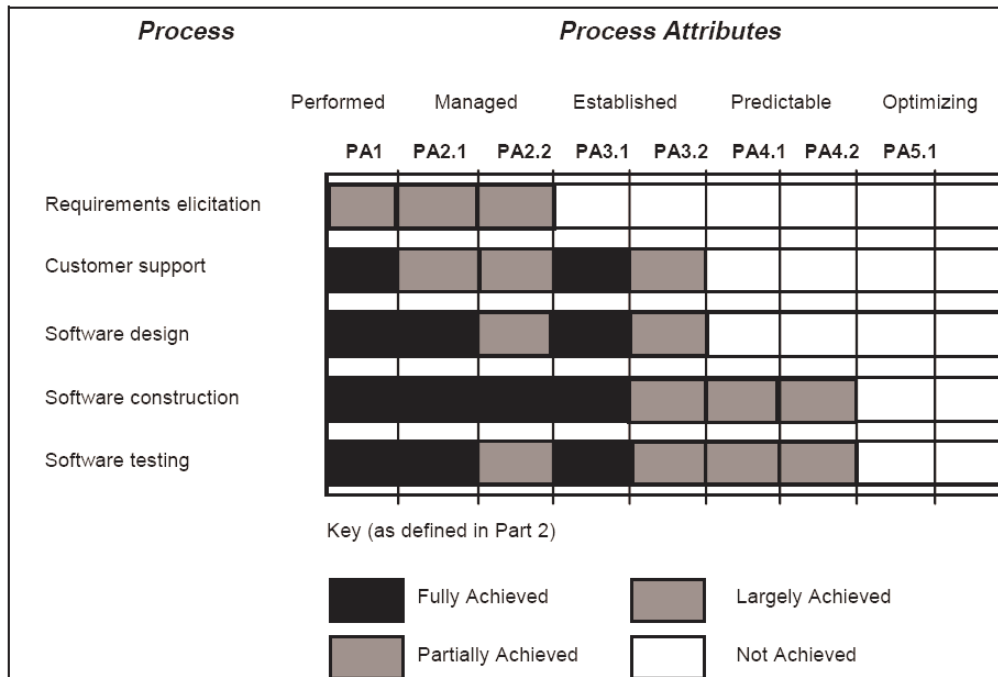
Francisco Ruiz - IS2

1.82



## Evaluación y Mejora de Procesos – ISO 15504

### • Resultados de la Evaluación (Assessed Capability)



Francisco Ruiz - IS2

1.83



## Evaluación y Mejora de Procesos – PSP Y TSP

- Tanto CMMI como ISO 15504 son marcos ideados para evaluar y mejorar a nivel de una **organización**.
- Pero existen otros dos niveles de mejora inferiores:
  - **Personal**
    - => **Personal Software Process - PSP**
  - **Equipo de Trabajo**
    - => **Team Software Process - TSP**
- Ambos se enmarcan en el contexto de una organización que aplica CMMI.

Francisco Ruiz - IS2

1.84



## Evaluación y Mejora de Procesos – PSP Y TSP

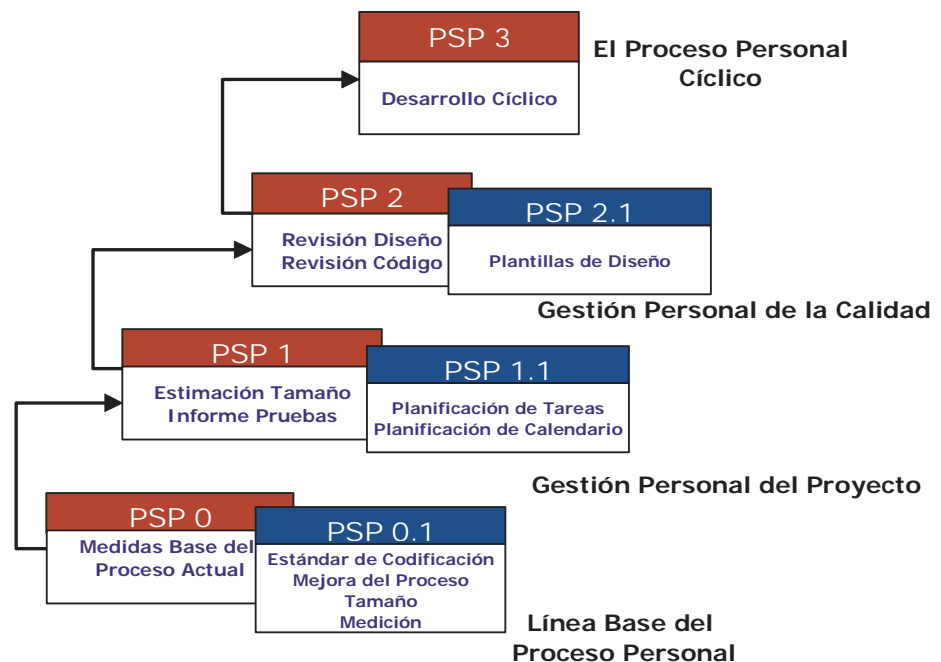
- **PSP (Personal Software Process)**

- Proporciona una serie de principios al ingeniero para llevar a cabo un proceso personal disciplinado.
  - Asiste a los ingenieros en la realización de planes precisos.
  - Determina los pasos que los ingenieros deben seguir para mejorar la calidad del producto.
  - Establece bancos de pruebas para medir la mejora del proceso personal.
  - Determina el impacto que los cambios del proceso tienen sobre el rendimiento del ingeniero.
- *Humprey, W.S. (2001): Introducción al Proceso Software Personal (PSP). Addison-Wesley.*



## Evaluación y Mejora de Procesos – PSP Y TSP

- **PSP** incluye 7 **procesos** a realizar por el ingeniero software.





## Evaluación y Mejora de Procesos – PSP Y TSP

- **PSP** utiliza tres **medidas** base:
  - Tiempo de desarrollo, defectos y tamaño.
  - Todas las demás medidas son derivadas de las anteriores.

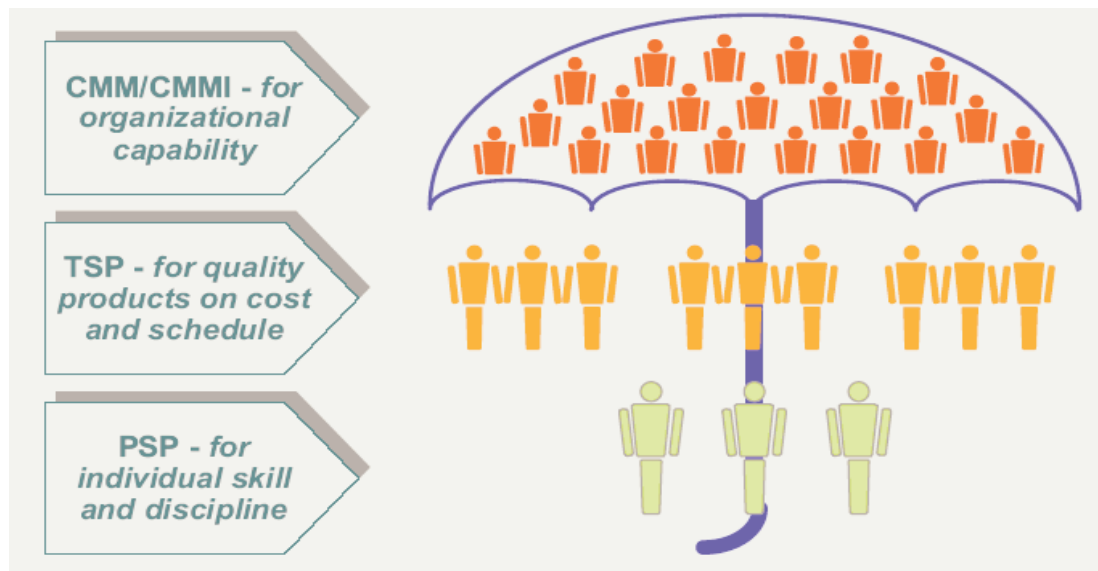
FORMULARIO DE REGISTRO DE TIEMPOS						
FECHA	COMIENZO	FIN	TIEMPO INTERRUPT.	T. DELTA	FASE	COMENTARIOS
13/5/05	7:58	8:45	3	44	Planificar	Llamada teléfono
	8:47	10:29	2	100	Diseñar	Crear y revisar diseño
	7:49	8:59		70	Codificar	Codificar <i>main</i> y todas las funciones
	9:15	9:46		31	Compilar	Compilar y enlazar
	9:47	10:10		23	Probar	Ejecutar pruebas A, B y C
	4:33	4:51		16	Postmortem	

*Ejemplo de formulario: registro de tiempos*



## Evaluación y Mejora de Procesos – PSP Y TSP

- Team Software Process (TSP)

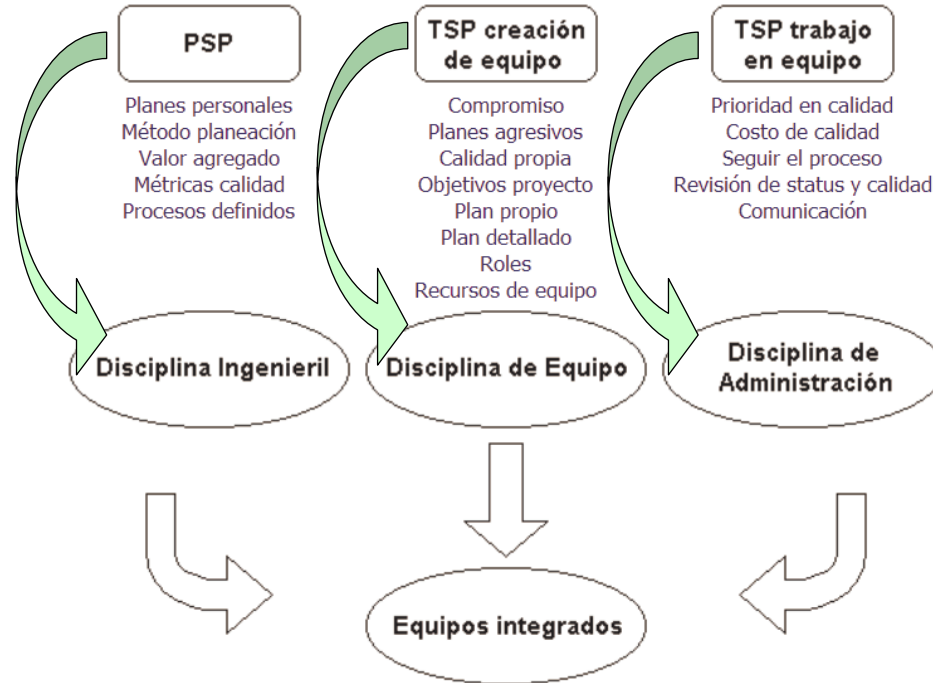


*Humprey, W.S. (2000): Introduction to the Team Software Process (TSP). Addison-Wesley.*



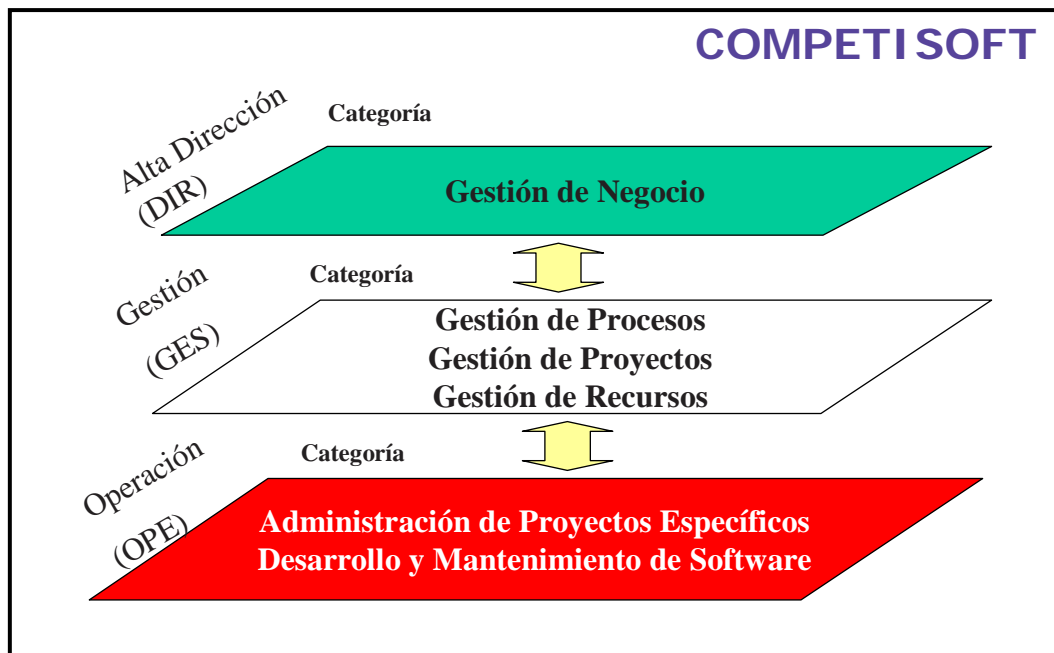
# Evaluación y Mejora de Procesos – PSP Y TSP

## • Estructura de TSP



# Evaluación y Mejora de Procesos

## • Existen otras propuestas aptas PYMES





## Medición del Software

*“Cuando puedas **medir** lo que estás diciendo y expresarlo en números, sabrás algo acerca de eso; pero cuando no puedes **medirlo**, cuando no puedes expresarlo en números, tus **conocimientos** serán escasos y no satisfactorios”*

Lord Kelvin

*“Lo que no sea **medible**, hazlo medible”*

Galileo Galilei

*“No se puede controlar lo que no se puede **medir**”*

Tom De Marco

*“No se puede predecir lo que no se puede **medir**”*

Norman Fenton

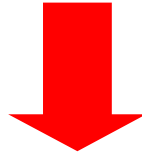


## Medición del Software

- Las **medidas** (o **métricas**) son un buen medio para entender, supervisar, controlar, predecir y probar durante los proyectos de desarrollo y mantenimiento de software.
- En general, la **medición de software** persigue tres **objetivos** fundamentales (Fenton y Pfleeger, 1997):
  - Entender qué ocurre durante el desarrollo y el mantenimiento.
  - Controlar qué es lo que ocurre en los proyectos.
  - Mejorar los procesos y los productos.



Las **medidas** pueden ser utilizadas para poder tomar las mejores decisiones (Pfleeger, 1997).



## MEDIDAS COMO MEDIO PARA ASEGURAR LA CALIDAD EN LOS PRODUCTOS/PROCESOS/ PROYECTOS SOFTWARE



- **Definiciones:** [SOFTWARE MEASUREMENT ONTOLOGY]
  - **Atributo**
    - Una propiedad mensurable, física o abstracta, que comparten todas las entidades de una cierta categoría de entidad.
      - Ej: Tamaño de código fuente.
  - **Medida (Métrica)**
    - Una forma de medir y una escala de medición.
      - La medida "líneas de código" puede ser definida para realizar mediciones del "tamaño" de un "módulo en C".
  - **Medida Base**
    - Una medida de un atributo que no depende de ninguna otra medida, y cuya forma de medir es un método de medición.
      - Ej: LCF (líneas de código fuente escritas), HPD (horas-programador diarias), CHP (coste por hora-programador, en unidades monetarias).



## Medición del Software

- **Definiciones:** [SOFTWARE MEASUREMENT ONTOLOGY]
  - **Medida Derivada**
    - Una medida que es derivada de otra medida base o derivada, utilizando una función de cálculo como forma de medir.
      - HPT (horas-programador totales, que es la sumatoria de las HPD de cada día).
  - **Indicador**
    - Una medida que es derivada de otras medidas utilizando un modelo de análisis como forma de medir.
      - PROD (productividad de los programadores), CAR (carestía del proyecto).
  - **Forma de Medir**
    - Secuencia de operaciones cuyo objeto es determinar el valor del resultado de la medición.
    - Una forma de medir puede ser un método de medición, una función de cálculo o un modelo de análisis.



## Medición del Software

- **Definiciones:** [SOFTWARE MEASUREMENT ONTOLOGY]
  - **Método de Medición**
    - La forma de medir una medida base.
    - Secuencia lógica de operaciones, descritas de forma genérica, usadas para realizar mediciones de un atributo respecto de una escala específica.
      - Contar líneas de código; anotar cada día las horas dedicadas por los programadores al proyecto.
  - **Función de Cálculo**
    - La forma de medir una medida derivada.
    - Algoritmo o cálculo realizado para combinar dos o más medidas base y/o derivadas.
      - $CTP = CHP * HPT$
  - **Modelo de Análisis**
    - La forma de medir un indicador.
    - Algoritmo o cálculo realizado para combinar una o más medidas (base, derivadas o indicadores) con criterios de decisión asociados.



## Definiciones: [SOFTWARE MEASUREMENT ONTOLOGY]

### Medición

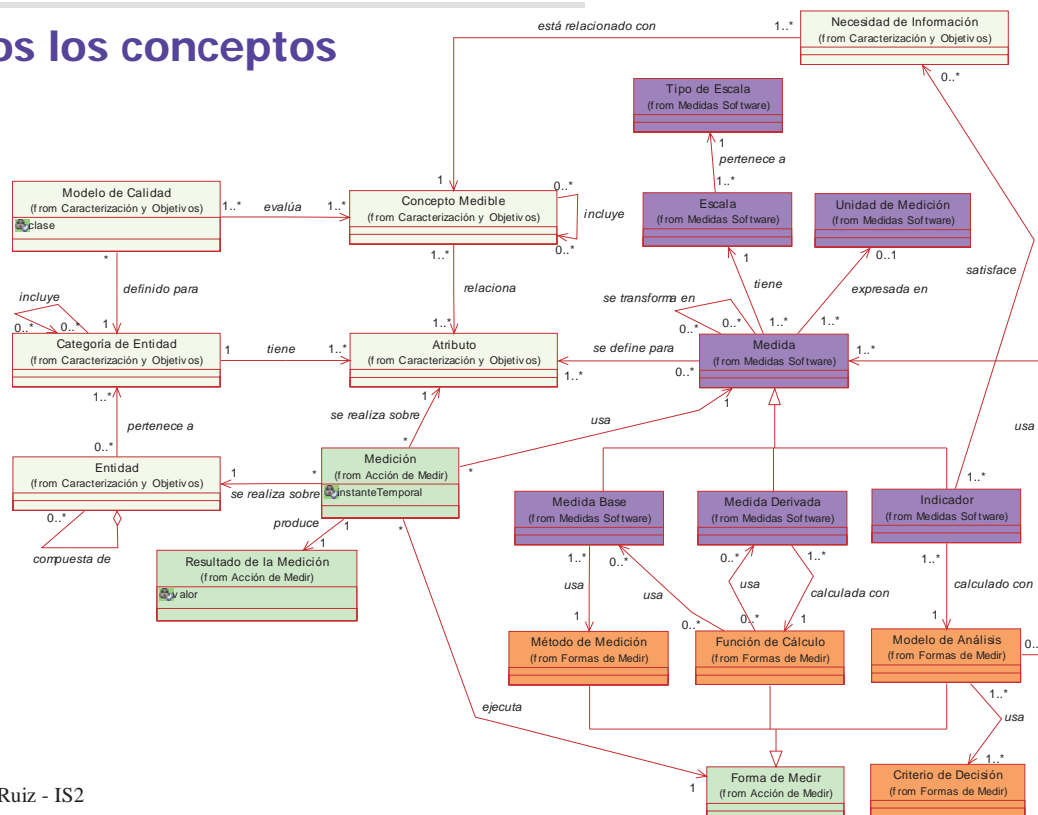
- Conjunto de operaciones que permite obtener el valor del resultado de la medición para un atributo de una entidad, usando una forma de medir.
  - Acción de usar la forma de medir "contar el número de líneas de código" para obtener el resultado de la medición del atributo "tamaño" de la entidad "módulo nominas.c".

### Resultado de la Medición

- Categoría o número asignado a un atributo de una entidad como resultado de una medición.
  - 35.000 líneas de código, 200 páginas, 50 clases, 5 meses desde el comienzo al fin del proyecto, 0.5 fallos por cada 1000 líneas de código.



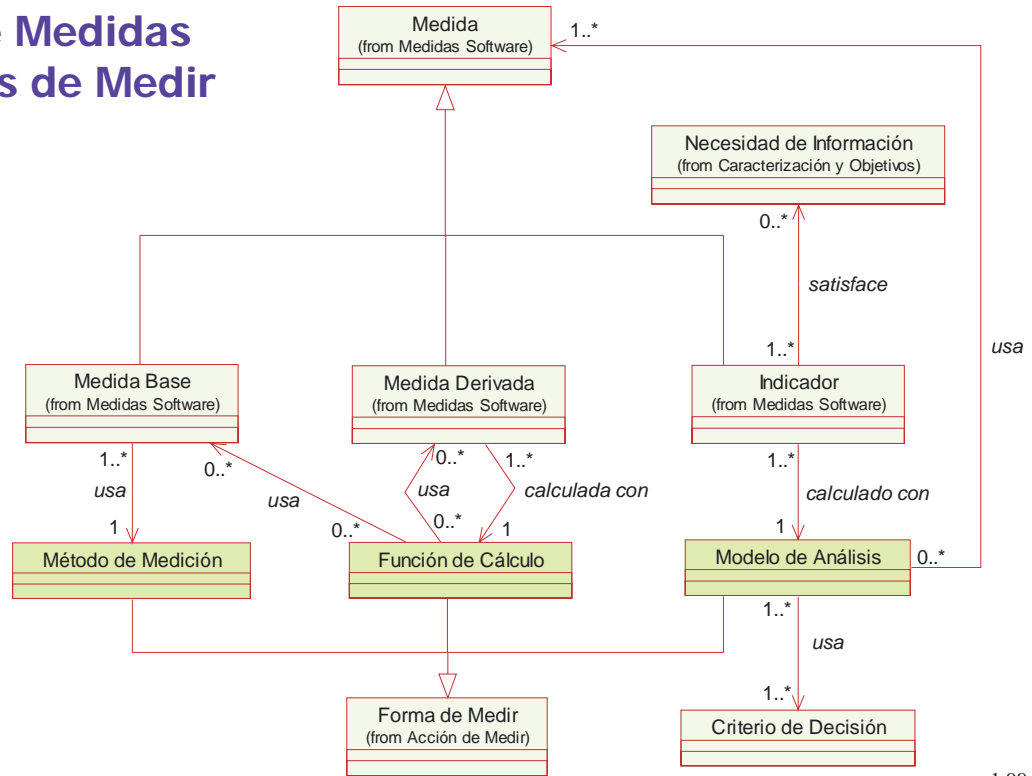
## Todos los conceptos





# Medición del Software

## Tipos de Medidas y Formas de Medir



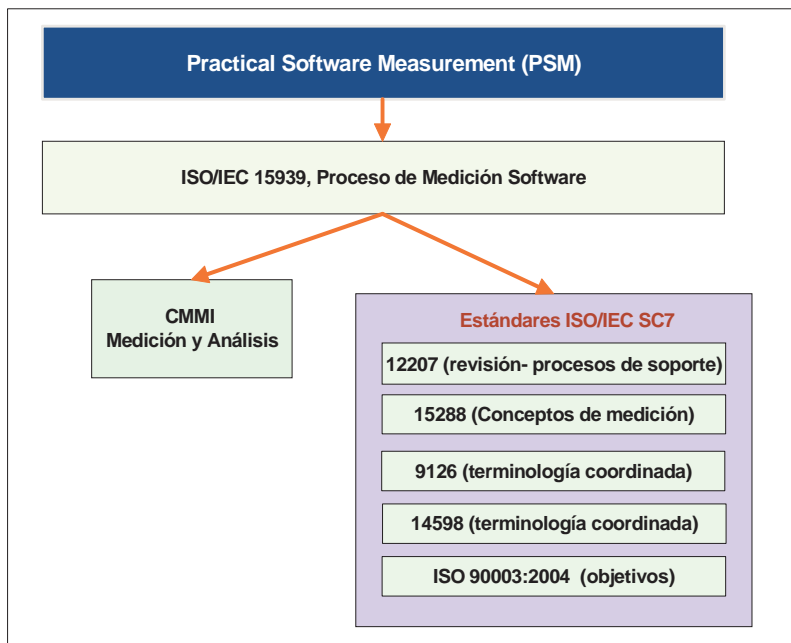
Francisco Ruiz - IS2

1.99



# Medición del Software - Proceso

## • Estándares y Metodologías:



GQM (Goal Question Metric)

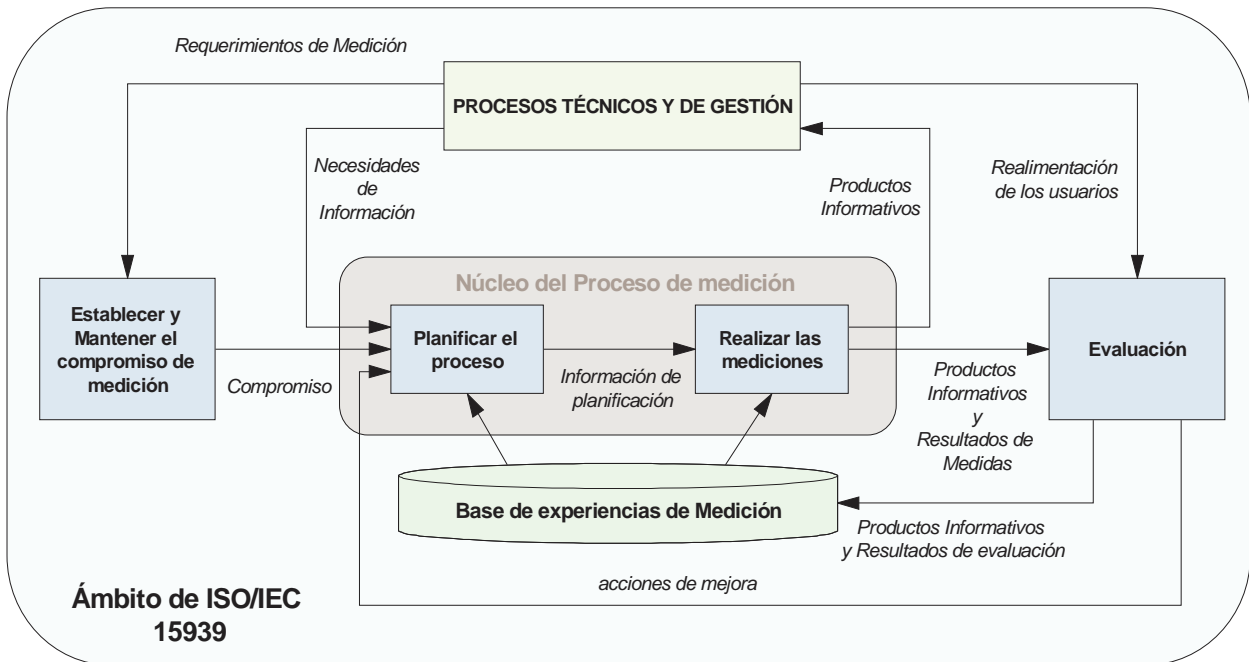
Francisco Ruiz - IS2

1.100



## Medición del Software - Proceso

- El **Proceso de Medición** según **ISO/IEC 15939**



## Medición del Software - GQM

- **Principio básico:** la medición debe ser realizada, siempre, orientada a un **objetivo**.
- **GQM** define un objetivo, refina este objetivo en preguntas y define medidas/métricas que intentan dar información para responder a estas preguntas.



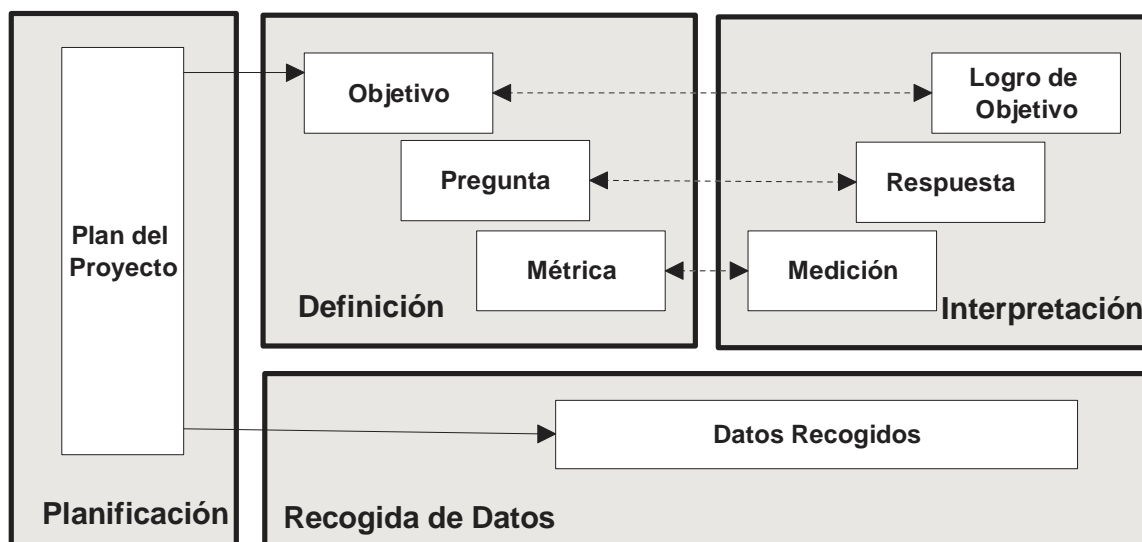
## Medición del Software - GQM

- El método **GQM** se lleva a cabo en las siguientes **fases**:
  - **Planificación**, se selecciona, define, caracteriza y planifica un proyecto de medición para la aplicación de la medición, obteniéndose como resultado un plan de proyecto.
  - **Definición**, se define y documenta el programa de la medición (objetivos, preguntas, medidas e hipótesis).
  - **Recopilación de Datos**, se recogen los datos reales de la medición.
  - **Interpretación**, se procesan los datos recopilados (resultados de medición) para obtener respuestas a las preguntas definidas, a partir de las cuales se puede evaluar el logro del objetivo planteado.



## Medición del Software - GQM

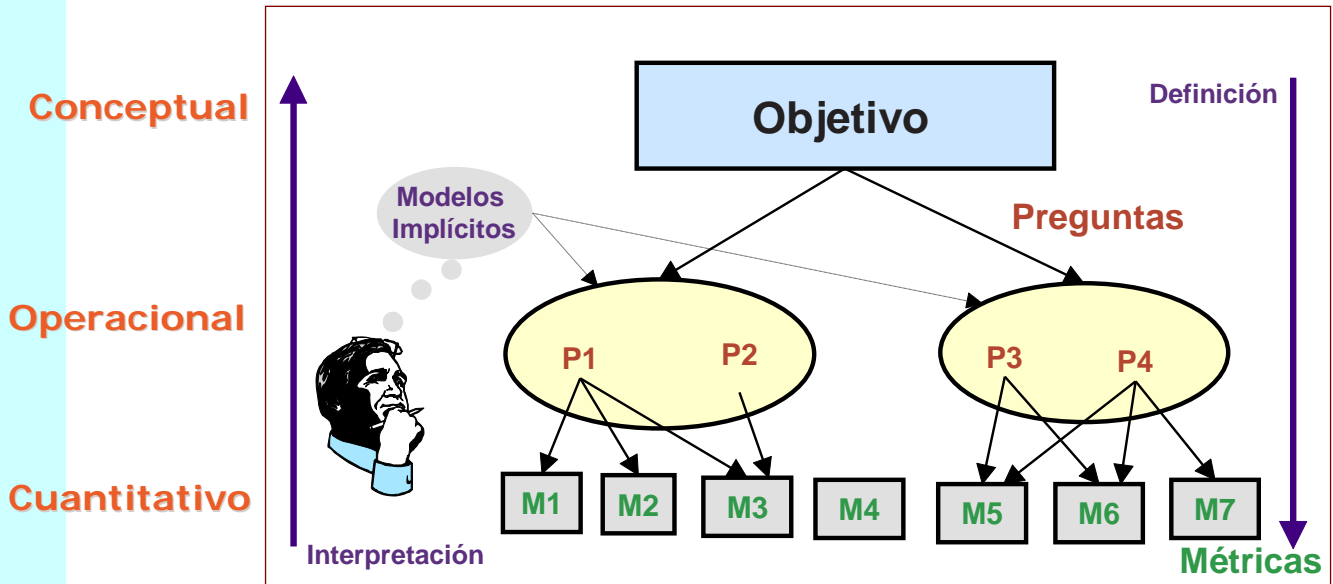
- **Fases GQM:**





## Medición del Software - GQM

- **Niveles de trabajo Goal-Question-Metric:**



## Medición del Software - GQM

- **Ejemplo** - Medida para BBDD Relacionales:

- **Objetivo GQM**

- *Analizar*
    - *Con el propósito de*
    - *Con respecto a*
    - *Desde el punto de vista de*
    - *En el contexto de*

**BBDD Relacionales**  
**Asegurar**  
**la Mantenibilidad**  
**los Diseñadores de BBDD**  
**Desarrollo y Mantenimiento de**  
**BBDD**

- **Preguntas:**

- **Pregunta 1.** ¿Cómo influye la complejidad de las tablas en la mantenibilidad de las bases de datos relacionales?
    - **Pregunta 2.** ¿Cómo influye la complejidad entre tablas en la mantenibilidad de las bases de datos relacionales?



# Medición del Software - GQM

## • Ejemplo - Medida para BBDD Relacionales:

### ■ Medidas:

#### ▪ Pregunta 1

- NA(T) - NÚMERO DE ATRIBUTOS DE UNA TABLA
- NFK(T) - NÚMERO DE CLAVES AJENAS
- RFK(T) - RATIO DE CLAVES AJENAS DE UNA TABLA

$$RFK(T) = \frac{NFK(T)}{NA(T)}$$

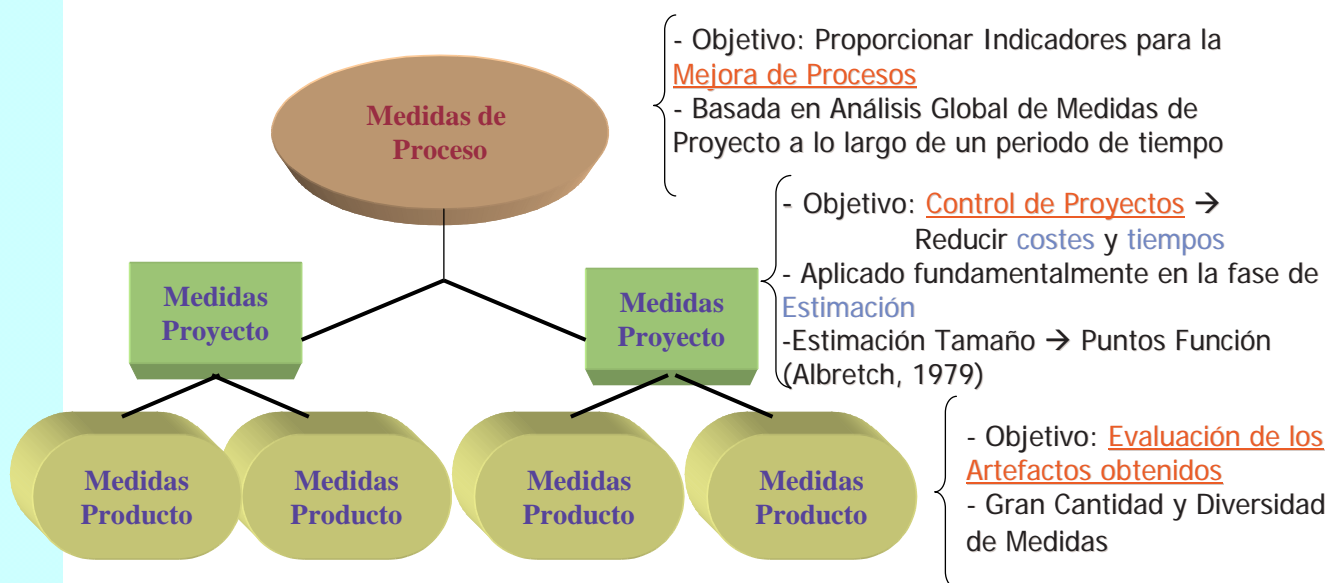
#### ▪ Pregunta 2

- NT - NÚMERO DE TABLAS
- NA - NÚMERO DE ATRIBUTOS
- NFK - NÚMERO DE CLAVES AJENAS (NFK)



# Medición del Software - Medidas

## • Tipos de Entidades y Medidas Software





# Medición del Software - Medidas

- Existen cientos de medidas/métricas software

Fase	Autores		Métricas
Análisis	Modelos Conceptuales	Albrecht (1973)	Punto Función
		DeMarco (1987)	Métrica Bang
Diseño	Estructurado	Card y Glass (1990)	Complejidad estructural, complejidad de datos, Complejidad del sistema
		Henry y Kafura (1981)	MHK
		Fenton (1991)	Tamaño, Profundidad, Anchura, Relación Arco-a-nodo
		(Biemman y Ott, 1994)	Métricas de Cohesión
		Dhama (1995)	Métricas de Acoplamiento
		McCabe (1976)	Métricas de Complejidad (Complejidad Ciclomática)
Modelos OO	Brito e Abreu v Carapuca (1994)	Métricas MOOD	
		Lorenz y K... (1994)	
Diseño Lógico Datos	Calero (200...)	Métricas Diagramas ER NE, NA, NDA, NCA, NMVA, NNR, NMNR, N1NR, NbinaryR, NN-AryR, NIS_AR, NreR, NRR)	
		DIT. Profundidad del Árbol de Herencia VOC. Número de Hijos. Métricas Modelos de Clases Nassoc, Nagg, Ndep, Ngen, NgenH, NaggH, MaxDIT, MaxHagg, NassosC, Hagg, NODP, NP, NW, Magg, NdepIn, NdepOut)	

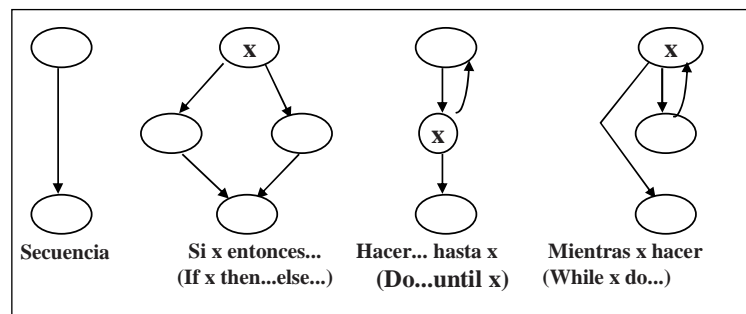
  

Fase	Autores		Métricas
Codificación		Halstead (1977)	Métricas Longitud Programa, Volumen Programa
		Martínez (2001)	Métricas Lenguajes 4GL (LDCp, NDA, V'(G), NSC, NCTI, NAVI, NSVM, LHSV, LDCe)
Pruebas		McCabe, Halstead	Métricas SQL (NCCT, NDCA, NSCT, NCCV, NSCV, NDS, NDI, NDD, NDU, NT, NA, A, NO, LDCs, LDCt)
		IEEE-982.1-1988 (1994)	Complejidad Ciclomática
Mantenimiento		Polo et al. (2002)	Profundidad de las Pruebas, Perfiles de Fallos
			IMS
			Métricas MANTEMA (TRCU, TRCNU, NPMCU, NPMCNU, MAXCU, MAXCNU, ...)



# Medición del Software - Medidas

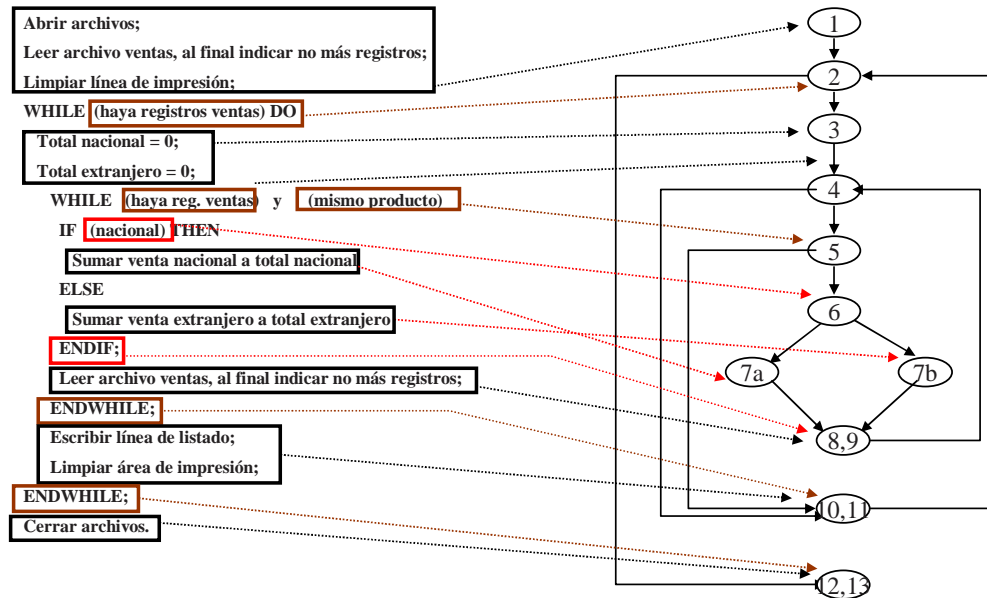
- Ejemplos de Medidas clásicas de producto:
  - LOC (Líneas de Código Fuente) [TAMAÑO]
  - Complejidad Ciclomática de McCabe [COMPLEJIDAD]
    - $V(G) = A - N + 2$ , siendo A el número de arcos del grafo y N el número de nodos.





# Medición del Software - Medidas

## Complejidad Ciclomática de McCabe



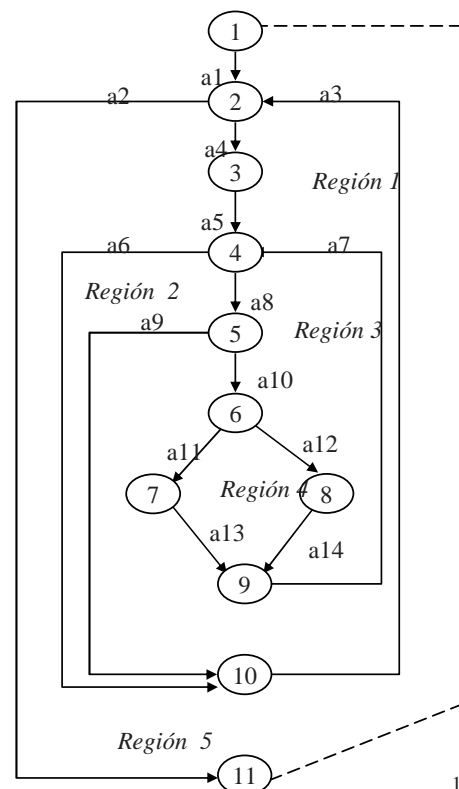
# Medición del Software - Medidas

## Complejidad Ciclomática (McCabe):

- Su valor proporciona el número de caminos independientes del conjunto básico de un programa.
- Mínimo número de casos de prueba necesarios para satisfacer criterios de Decisiones

### Caminos Independientes:

- 1-2-11 (a1, a2)
- 1-2-3-4-10-2-11 (a1, a4, a5, a6, a3, a2)
- 1-2-3-4-5-10-2-11 (a1, a4, a5, a8, a9, a3, a2)
- 1-2-3-4-5-6-7-9-4-10-2-11 (a1, a4, a5, a8, a10, a11, a13, a7, a6, a3, a2)
- 1-2-3-4-5-6-8-9-4-10-2-11 (a1, a4, a5, a8, a10, a12, a14, a7, a6, a3, a2)





# Medición del Software - Medidas

## Medidas para Sistemas OO:

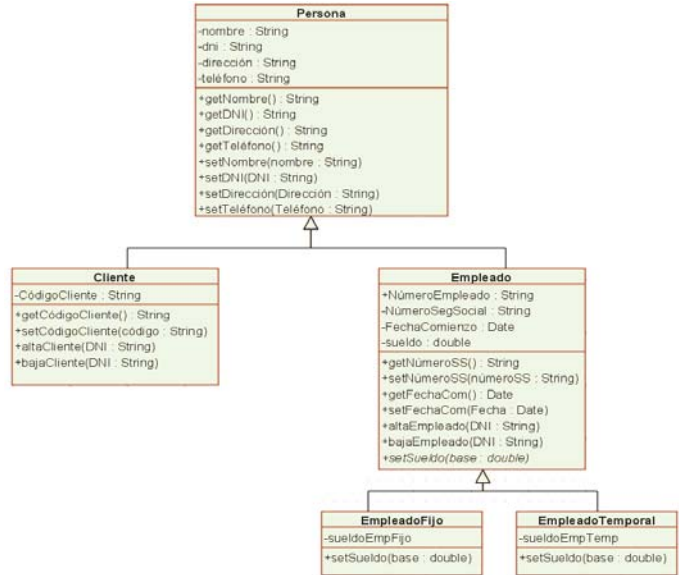
- Métodos Ponderados por Clase (WMC)
- Profundidad del Árbol de Herencia de una Clase (DIT)
- Número de Hijos (NOC)

$$WMC = \sum_{i=1}^n C_i$$

**WMC(Persona) = 8**

**DIT(Persona) = 0**  
**DIT(Empleado Fijo)=2**

**NOC(Persona) = 2**  
**NOC (Empleado) =2**



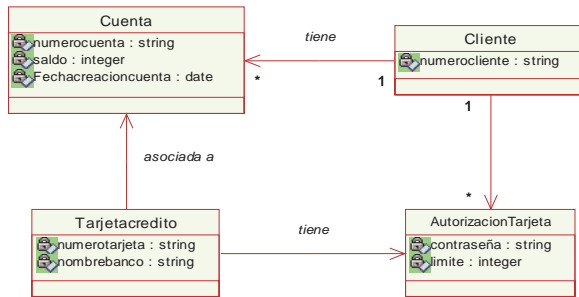
Francisco Ruiz - IS2



# Medición del Software - Medidas

## Medidas para Sistemas OO:

- Acoplamiento entre Objetos (CBO)



**CBO(Cuenta) = 0**  
**CBO(Cliente) = 2**

- Respuesta para una Clase (RFC)

**RFC(A)=10**

Clase A con cuatro métodos:  
 A::f1() invoca B::f1(), B::f2() y C::f3()  
 A::f2() invoca B::f1()  
 A::f3() invoca A::f4(), B::f3(), C::f1() y C::f2()  
 A::f4() No llama a otros métodos

Entonces

$$RS = \{ A::f1, A::f2, A::f3, A::f4 \} \cup \{ B::f1, B::f2, C::f3 \} \cup \{ B::f1 \} \cup \{ A::f4, B::f3, C::f1, C::f2 \} = \{ A::f1, A::f2, A::f3, A::f4, B::f1, B::f2, B::f3, C::f1, C::f2, C::f3 \}$$

Francisco Ruiz - IS2

1.114



- **Software Quality Management (SQM)**
  - Es **aplicable a** los productos, procesos y recursos.
  - Los **procesos** SQM tienen como objetivos:
    - Conseguir que los productos software funcionan bien,
    - Satisfacer los requisitos de clientes y demás *stakeholders*,
    - Proveer valor a clientes y demás *stakeholders*,
    - Proveer la calidad del software necesaria para satisfacer los requisitos software.
  - SQM puede usarse para evaluar tanto productos intermedios como el producto final.



- Los principales **procesos SQM** son:
  - **Aseguramiento de Calidad** (*Quality Assurance*)
  - **Verificación** (*Verification*)
  - **Validación** (*Validation*)
  - **Revisión** (Review)
  - **Auditoría** (*Audit*)



## Gestión de la Calidad - Aseguramiento

- **Software Quality Assurance (SQA)**
  - **Busca** asegurar que los productos y procesos software son conformes con los requisitos
  - **mediante** la realización de actividades que dan confianza en que la calidad está siendo incorporada en el software:
    - El problema es clara y adecuadamente establecido.
    - Los requisitos de la solución son se definen y expresan de forma correcta.
    - Los problemas se identifican lo antes posible.
    - Asegurar que los procesos son apropiados y después se eimplementa según lo planificado.



## Gestión de la Calidad - Aseguramiento

- El **Plan SQA** define los medios que se usarán para asegurar que el software desarrollado satisface los requisitos de usuario y es de la calidad más alta posible dentro de las restricciones del proyecto.
  - Debe ser consistente con la **gestión de la configuración** del software.
  - Identifica documentos, estándares, prácticas y convenciones aplicables en el proyecto y cómo éste será controlado y supervisado.
  - También establece medidas, estadísticas, procedimientos para informar de problemas, acciones correctivas, recursos (herramientas), ..



- **Verificación y Validación (VV)** es un conjunto de procedimientos, actividades, técnicas y herramientas que se utilizan, paralelamente al desarrollo de software, para **asegurar que un producto software resuelve el problema inicialmente planteado**.
- Las pruebas son una familia de técnicas de **VV**
- Principio:
  - Actuar sobre los productos intermedios que se generan durante el desarrollo para detectar y corregir cuanto antes sus defectos y las desviaciones respecto al objetivo fijado.



- Los **objetivos concretos de la VV** son:
  - Detectar y corregir los **defectos** tan **pronto** como sea posible en el ciclo de vida del software.
  - Disminuir los **riesgos**, las desviaciones sobre los presupuestos y sobre el calendario o programa de tiempos del proyecto.
  - Mejorar la **calidad y fiabilidad** del software.
  - Mejorar la **visibilidad** de la **gestión** del proceso de desarrollo.
  - **Valorar** rápidamente los **cambios** propuestos y sus consecuencias.



## Verificación y Validación

[repaso tema 5]

- La visión del desarrollo de software como un conjunto de fases con posibles realimentaciones facilita la **VV**.
  - Al inicio del proyecto es necesario hacer un [Plan de VV del SW](#) (IEEE 1012).
  - Las actividades de VV se realizan de forma iterativo durante el desarrollo.
  - IEEE 1012-2004: IEEE **Standard** for Software Verification and Validation.

1. Propósito
2. Documentos de referencia
3. Definiciones
4. Visión general de la verificación y validación
  - 4.1 Organización
  - 4.2 Programa de tiempos
  - 4.3 Esquema de integridad de software
  - 4.4 Resumen de recursos
  - 4.5 Responsabilidades
  - 4.6 Herramientas, técnicas y metodologías
5. Verificación y validación en el ciclo de vida
  - 5.1 Gestión de la VV
  - 5.2 VV en el proceso de adquisición
  - 5.3 VV en el proceso de suministro
  - 5.4 VV en el proceso de desarrollo:
    - 5.4.1 VV de la fase de concepto
    - 5.4.2 VV de la fase de requisitos
    - 5.4.3 VV de la fase de diseño
    - 5.4.4 VV de la fase de implementación
    - 5.4.5 VV de la fase de pruebas
    - 5.4.6 VV de la fase de instalación
  - 5.5 VV de la fase de operación
  - 5.6 VV del mantenimiento
6. Informes de la VV del software
7. Procedimientos administrativos de la VV
  - 7.1 Informe y resolución de anomalías
  - 7.2 Política de iteración de tareas
  - 7.3 Política de desviación
  - 7.4 Procedimientos de control
  - 7.5 Estándares, prácticas y convenciones.
8. Requisitos de documentación para la VV



## Verificación y Validación

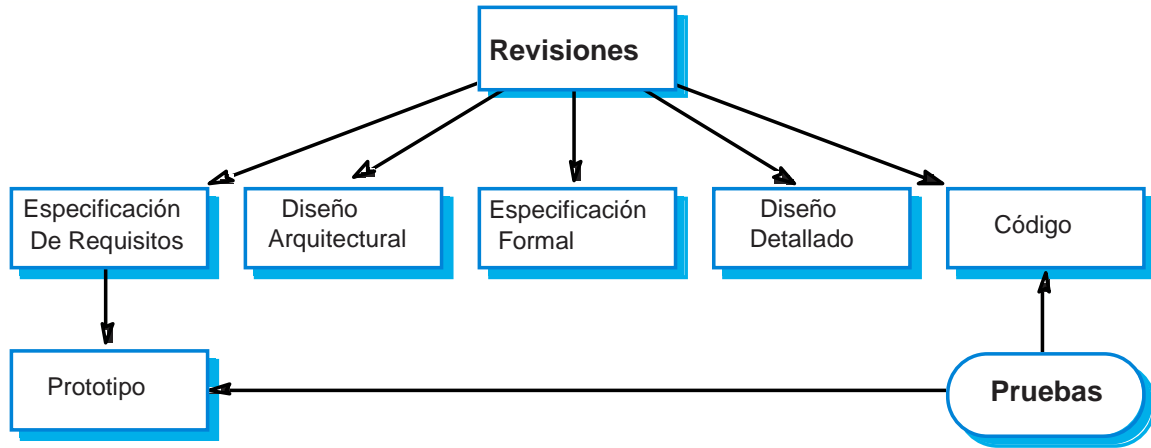
[repaso tema 5]

- **Actividades de VV**
  - **Verificación:**
    - ¿Estamos **construyendo correctamente** el producto?
    - El software debe ser conforme a su especificación.
    - Objetivo: Demostrar la consistencia, compleción y corrección de los **artefectos software** entre las **fases** del ciclo de desarrollo de un proyecto.
    - Técnica más utilizada: **Revisiones SW**
  - **Validación:**
    - ¿Estamos construyendo el **producto correcto**?
    - El software debe hacer lo que el usuario realmente quiere
    - Objetivo: Determinar la corrección del **producto final** respecto a las necesidades del usuario.
    - Técnica más utilizada: **Pruebas SW**



• **VV del Código:**

- **Técnicas Dinámicas (Pruebas) vs Estáticas (Inspecciones)**



• **Tipos de Revisiones y Auditorías:**

OBJETIVOS DE CALIDAD	MÉTODO PRINCIPAL
EVALUACIÓN	REVISIÓN DE GESTIÓN REVISIÓN TÉCNICA
VERIFICACIÓN	INSPECCIÓN WALK-THROUGH (RECORRIDO)
VALIDACIÓN	PRUEBAS
CONFIRMACIÓN DEL CUMPLIMIENTO DE NORMAS Y PROCEDIMIENTOS	AUDITORÍA



# Verificación y Validación

## • Actividades de Verificación

- Algunas Actividades de Verificación en el Ciclo de Vida

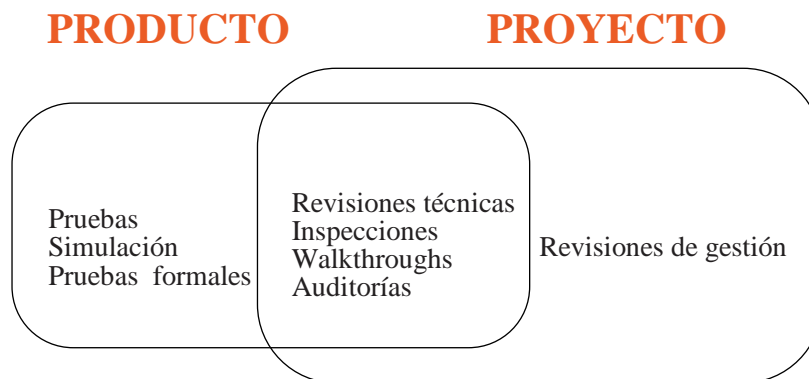
FASE DEL CICLO DE VIDA	ACTIVIDADES DE VERIFICACIÓN
<b>REQUISITOS</b>	<ol style="list-style-type: none"> <li>1. Determinar el enfoque de verificación</li> <li>2. Comprobar adecuación de los requisitos</li> <li>3. Generar datos de prueba funcionales</li> </ol>
<b>DISEÑO</b>	<ol style="list-style-type: none"> <li>1. Determinar la consistencia del diseño con los requisitos</li> <li>2. Determinar la adecuación del diseño</li> <li>3. Generar datos de prueba estructurales y funcionales</li> </ol>
<b>CONSTRUCCIÓN</b>	<ol style="list-style-type: none"> <li>1. Determinar la consistencia respecto del diseño</li> <li>2. Determinar la adecuación de la implementación</li> <li>3. Generar datos de prueba estructurales y funcionales</li> <li>4. Aplicar los datos de prueba</li> </ol>
<b>OPERACIÓN Y MANTENIMIENTO</b>	<ol style="list-style-type: none"> <li>1. Nueva ejecución de pruebas</li> </ol>



# Verificación y Validación

## • Revisiones y Auditorías:

- Relación con algunos procesos de aseguramiento de calidad





## Verificación y Validación – Revisiones

- **Revisiones de Gestión:**

- Estudiar el progreso del proyecto y la realización de actividades según el plan
- Adecuación del enfoque de gestión del proyecto para lograr sus objetivos.
- Ayudar a las decisiones de cambios de gestión en el proyecto
- Confirmar los requisitos y su asignación en el sistema.

- **Revisiones Técnicas.**

- El producto se ajusta a sus especificaciones.
- El desarrollo (o mantenimiento) del producto intermedio se está realizando de acuerdo a los planes, estándares y guías aplicables al proyecto.
- Los cambios en el producto se realizan adecuadamente y afectan sólo a aquellas áreas identificadas por la especificación de cambios.



## Verificación y Validación – Revisiones

- **Inspecciones** → Producto intermedio.

- Verificar si el producto satisface sus especificaciones o atributos de calidad
- Verificar si el producto se ajusta a los estándares utilizados en la empresa.
- Señalar las desviaciones sobre los estándares y las especificaciones.
- Recopilar datos que realimenten inspecciones posteriores (defectos recogidos, esfuerzo empleado, etc.) y ayudar a su utilización práctica.
- No pretende examinar alternativas o aspectos de estilo.

- **Walkthroughs** [Recorridos]

- Buscar defectos, omisiones y contradicciones.
- Mejorar el producto.
- Evaluar conformidad con estándares o normas.
- Considerar posibles soluciones y alternativas a los problemas encontrados.



## Verificación y Validación – Revisiones

- Otra clasificación de las revisiones
  - **Revisiones informales:**
    - No hay procedimientos definidos, por lo que la revisión se realiza de la forma más flexible posible.
    - Ventajas → menor coste y esfuerzo, preparación corta, etc.
    - Desventajas → Detectan menos defectos
  - **Revisiones semi-formales:** Se definen unos procedimientos mínimos a seguir → *walkthroughs*.
  - **Revisiones formales:**
    - Se define completamente el proceso, los participantes y sus funciones, los documentos, etc.
    - **Inspecciones**



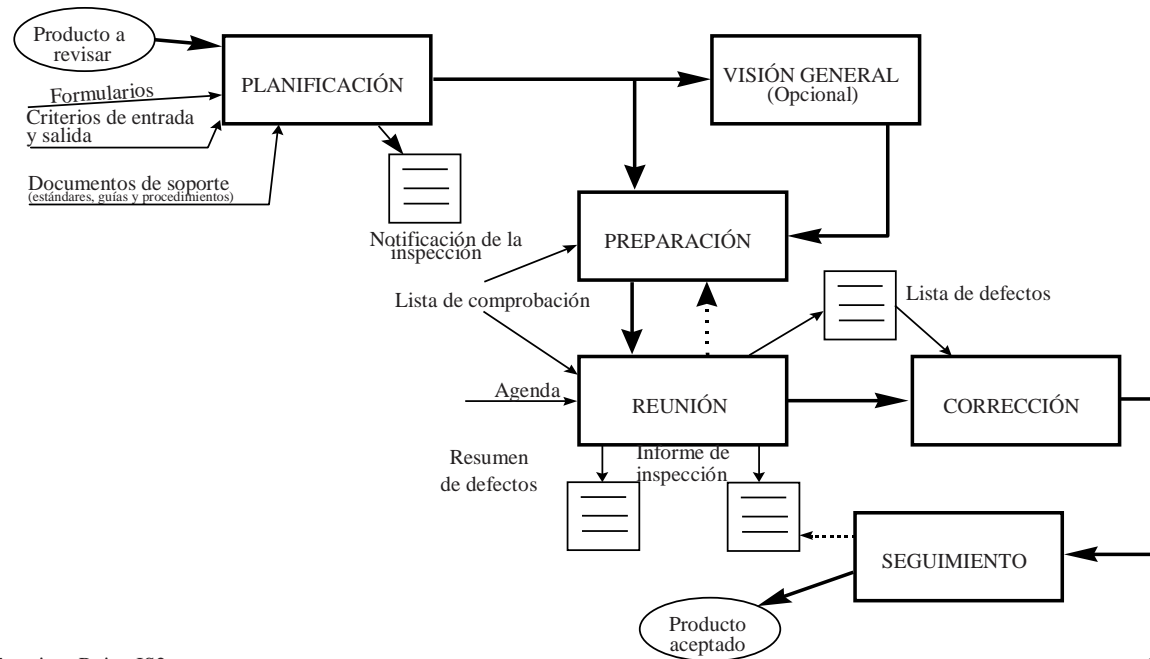
## Verificación y Validación - Inspecciones

- **Inspecciones**
  - El objetivo es detectar y registrar los **defectos de un producto intermedio**.
  - Consisten en
    - Verificar si el producto satisface sus **especificaciones o atributos de calidad**
    - Verificar si el producto se ajusta a los **estándares** utilizados en la empresa.
    - Señalar las **desviaciones** sobre los estándares y las especificaciones.
    - **Recopilar datos** que realimenten inspecciones posteriores (defectos recogidos, esfuerzo empleado, etc.) y ayudar a su utilización práctica.
    - No pretende examinar alternativas o aspectos de estilo.



## Verificación y Validación - Inspecciones

- Flujo de Trabajo para realizar una **Inspección**



Francisco Ruiz - IS2

1.131



## Verificación y Validación - Inspecciones

- Roles Involucrados:**

- Jefe del Proyecto
  - Responsable de las actividades administrativas.
- Coordinador
  - Elegido por el jefe de proyecto para el aseguramiento de la calidad.
- Moderador
  - Comprueba que se siguen los procedimientos de la inspección.
- Autor
  - Responsable de corregir los errores que se detecten.
- Lector
  - Guía al resto del equipo.
- Inspector
  - Detecta defectos.
- Secretario
  - Anota y clasifica los defectos encontrados.

Francisco Ruiz - IS2

1.132



## Verificación y Validación - Inspecciones

- Participación de los Roles en las Tareas

ETAPA	Moderador	Autor	Lector	Secretario	Inspector
<b>Planificación</b>	X	X			
<b>Visión General</b>	X	X	-	-	-
<b>Preparación</b>	X	X	X	X	X
<b>Reunión</b>	X	X	X	X	X
<b>Corrección</b>		X			
<b>Seguimiento</b>	X	X	(X)	(X)	(X)



## Verificación y Validación - Inspecciones

- Etapas (i)

ETAPAS DE LA INSPECCIÓN			
ETAPA	OBJETIVOS	TAREAS	PARTICIPACIÓN
<b>PLANIFICACIÓN</b>	Organización.	Aprobar los criterios de entrada. Establecer el programa de tiempos. Elegir los participantes. Determinar realización o no de presentación. Preparar la notificación de inspección. Distribuir el material a los inspectores.	Moderador. Autor.
<b>VISIÓN GENERAL</b>	Enseñar el producto.	Presentación del producto.	Todos los inspectores.
<b>PREPARACIÓN</b>	Comprensión del producto. Identificar defectos.	Estudio individual del producto.	Todos los inspectores.



## Verificación y Validación - Inspecciones

- Etapas (ii)

ETAPAS DE LA INSPECCIÓN			
ETAPA	OBJETIVOS	TAREAS	PARTICIPACIÓN
<b>REUNIÓN</b>	Verificar el producto.	Introducción a la reunión. Comprobar la preparación de los inspectores. Lectura y detección de defectos. Revisión de la lista de defectos. Determinar la disposición del producto.	Todos los inspectores.
<b>TERCERA HORA</b>	No perder posibles soluciones.	Analizar posibles soluciones o mejoras.	Todos los inspectores.
<b>CORRECCIÓN</b>	Cumplir los criterios de salida.	Resolución de todos los defectos.	Autor.
<b>SEGUIMIENTO</b>	Verificar la corrección.	Comprobar que todos los defectos están resueltos y que no se añaden otros nuevos.	Moderador.
<b>ANÁLISIS DE RESULTADOS</b>	Analizar los resultados para inspecciones posteriores.	Clasificar los defectos por clase, gravedad y tipo.	Coordinador.



## Verificación y Validación - Inspecciones

- Inspecciones. [Informes de la Inspección](#)
  - Proporcionan información del progreso o resultados de la inspección
  - Estructura:
    - Notificación de la reunión de inspección → Anuncio formal
    - Lista de Defectos → Registro detallado de cada defecto descubierto
      - Localización
      - Descripción → Problema
      - Clasificación
    - Informe Resumen de Defectos
    - Informe de Inspección



## Verificación y Validación - Inspecciones

- Inspecciones. [Informes de la Inspección](#)
  - Lista de Defectos
    - Clasificación: Factor de Calidad, Tipo, Clase, Gravedad

TIPO DE DEFECTO	DESCRIPCIÓN
<b>Cumplimiento de estándares</b>	Desviación sobre los estándares que debe seguir el producto.
<b>Factores humanos</b>	Procedimientos operativos incorrectos.
<b>Documentación</b>	Descripciones inadecuadas de algún componente (por ejemplo, comentarios incorrectos).
<b>Funcionalidad</b>	Defectos en la especificación de las funciones de un componente
<b>Interfaz</b>	Defectos en la comunicación entre componentes del software (por ejemplo, llamadas incorrectas de los módulos, paso de datos incorrectos entre módulos).
<b>Datos</b>	Defectos en la especificación de los datos (por ejemplo, declaraciones, inicializaciones o descripciones de datos incorrectas).



## Verificación y Validación - Inspecciones

- [Informes de Inspección](#)
  - Lista de Defectos clasificados por Factor de Calidad, Tipo, Clase, o Gravedad

TIPO DE DEFECTO	DESCRIPCIÓN
<b>Lógico</b>	Defectos en la lógica de control de un módulo (por ejemplo, límites de los bucles incorrectos).
<b>Entrada/Salida</b>	Defectos en la comunicación con dispositivos.
<b>Sintaxis</b>	Defectos gramaticales.
<b>Casos de prueba</b>	Especificaciones incompletas de una condición de prueba o una desviación del plan de pruebas.
<b>Entorno de pruebas</b>	Defectos en la definición o especificación software o hardware de pruebas, nivel de seguridad, etc.
<b>Plan de pruebas</b>	Defectos en la definición o especificación del alcance de las pruebas.
<b>Ejecución</b>	Falta de la eficiencia de ejecución prevista.



# Verificación y Validación - Inspecciones

- Informes de Inspección
  - Ejemplo de Formulario

Informe de inspección								
Proyecto	Elemento revisado							
Fecha de revisión	Versión							
Documento								
Moderador								
Revisores								
Tipo de reunión: <input type="checkbox"/> Revisión <input type="checkbox"/> Re-revisión <input type="checkbox"/> Mantenimiento								
Tipo de inspección:								
<input type="checkbox"/> Requisitos (ARS)	<input type="checkbox"/> Especificación (EFS)	<input type="checkbox"/> Arquitectura (DTS)						
<input type="checkbox"/> Diseño (DTS)	<input type="checkbox"/> Código (DCS)	<input type="checkbox"/> Diseño de Pruebas (DTS)						
<input type="checkbox"/> Casos de prueba (DCS)								
Decisión: <input type="checkbox"/> Aceptar <input type="checkbox"/> Aceptar condicionalmente <input type="checkbox"/> Rechazar								
Tipo	Defectos graves				Defectos leves			
	Omisión	Error	Añadido	Total	Omisión	Error	Añadido	Total
Sintaxis								
Compleción								
Nombres								
Consistencia								
Esfuerzo:								
- Preparación				horas-persona				
- Reuniones				horas-persona				
- Seguimiento				horas-persona				
Firma:								
Moderador				Secretario				
D/Dña. _____				D/Dña. _____				

Francisco Ruiz - IS2

1.139



# Verificación y Validación - Inspecciones

- Listas de Comprobación (*checklist*)
  - Conjunto de Preguntas (respuesta Sí/No), confeccionadas por expertos, sobre las características que deberá cumplir el producto

LISTA DE COMPROBACIÓN PARA LA REVISIÓN DE UNA ERS
<b>Organización y completión</b>
¿Son correctas las referencias cruzadas a otros requisitos?
¿Están todos los requisitos escritos de una manera consistente y a un nivel apropiado de detalle?
...
<b>Corrección</b>
¿Hay algún requisito que entre en conflicto o duplique a otro requisito?
¿Está cada requisito escrito de una forma clara, concisa y no ambigua?
¿Se puede verificar cada requisito?
...
<b>Atributos de Calidad</b>
¿Están especificados correctamente todos los requisitos de rendimiento?
¿Están correctamente especificadas todas las consideraciones de seguridad?
<b>Trazabilidad</b>
¿Está cada requisito identificado de una manera única y correcta?
¿Es trazable cada requisito funcional a otros requisitos de más alto nivel?
<b>Otros aspectos</b>
¿No se incluyen como requisitos funcionales soluciones de diseño o implementación?
¿Están identificadas las funciones críticas en tiempo y especificados los criterios de tiempo de las mismas?

LISTA DE COMPROBACIÓN PARA LA REVISIÓN DE UN MODELO DE CASOS DE USO
<b>Descomposición en paquetes</b>
¿Está definido un nombre único por cada paquete y existe una descripción breve?
El nombre del paquete ¿es claro y fácilmente entendible?
...
<b>Actores</b>
¿Están definidos todos los actores del sistema?
¿Es correcta la definición de cada actor (cada actor define un único rol)?
...
<b>Casos de Uso</b>
¿Están definidos todos los casos de uso del sistema?
¿Hay casos de uso repetidos o de funcionalidad muy parecida?
El nombre del caso de uso ¿es claro y fácilmente comprensible?
<b>Diagramas de Casos de Uso</b>
Por cada paquete ¿existe al menos un diagrama de casos de uso?
Los diagrama de casos de uso ¿tienen definida una descripción clara?
¿Son correctas las relaciones entre casos de uso ( <i>uses, extends</i> )?
¿Son correctas las relaciones de los actores con los casos de uso?

Francisco Ruiz - IS2

1.140



## Verificación y Validación - Recorridos

- Walkthrough:

	INSPECCIÓN	WALKTHROUGH
<b>Objetivo</b>	Detección de defectos.	Detección de defectos. Comentarios sobre el estilo. Búsqueda de soluciones. Intercambio de conocimientos.
<b>Formalidad</b>	Formal.	Informal o Semiformal.
<b>Composición del equipo</b>	Personas de distinto nivel jerárquico, que pueden pertenecer además a otros proyectos.	Personas del mismo nivel del equipo de desarrollo.
<b>Papeles definidos de los participantes</b>	Sí.	No.
<b>Utilización de listas de comprobación</b>	Siempre.	A veces.
<b>Clasificación de defectos</b>	Sí.	No.
<b>Análisis de resultados para realimentar nuevas revisiones</b>	Sí.	No.

Francisco Ruiz - IS2

1.141



## Verificación y Validación - Auditorías

- Auditorías:

- **Auditoría funcional (AFU):**

- Es un examen sobre el software justo antes de su entrega → Verificar que cumple todos los requisitos definidos en la ERS.

- **Auditoría física (AFI):**

- Es un examen que se realiza para verificar que el software y su documentación son consistentes y están preparados para su entrega.

- **Auditoría durante el proceso de desarrollo (AP):**

- Se realiza para verificar la consistencia del diseño, que incluye el análisis de:
  - Código frente a la documentación de diseño.
  - Especificaciones de interfaz (software y hardware).
  - Implementaciones de diseño frente a los requisitos funcionales.
  - Requisitos funcionales frente a las descripciones de pruebas.

Francisco Ruiz - IS2

1.142



## Verificación y Validación - Auditorías

- Auditorías vs Revisiones:

ATRIBUTO	REVISIONES	AUDITORÍAS
MECANISMO	Las reuniones.	Mezcla de reuniones, observaciones y exámenes.
RESPONSABILIDAD	Generalmente compartida entre un grupo de personas pertenecientes a la organización.	Realizada por un grupo personas que no suelen pertenecer a la organización, en el que sobresale la figura central del "auditor".
DURACIÓN	Corta: unas pocas horas.	De media a larga: de días a meses.
ANIDAMIENTO	Las reuniones pueden tener múltiples sesiones.	Puede incluir otras auditorías, revisiones, e incluso, algunas pruebas periódicas.
FRECUENCIA	Depende de la fase del ciclo de vida.	Periódica.



## Verificación y Validación – Otras Técnicas

- Otras Técnicas de Verificación y Validación

- Análisis de algoritmos**

- Verificar la funcionalidad de los algoritmos y recoger estadísticas sobre el consumo de recursos en tiempo de ejecución.

- Análisis de simulación**

- Proporcionar una evaluación del rendimiento y la información necesaria para planificar la capacidad de un sistema durante su diseño.

- Audidores de código**

- Examinar el código fuente y determinar automáticamente si se siguen los estándares y prácticas de programación descritos previamente.



## Verificación y Validación – Otras Técnicas

- Otras Técnicas de Verificación y Validación
  - **Generadores de referencias cruzadas**
    - Producir listas de nombres de variables, procedimientos, etiquetas, etc. determinando su ubicación dentro de un programa.
  - **Analizadores de flujo de control**
    - Determinar la presencia o ausencia de errores del flujo de control, es decir, secuencias incorrectas en la ejecución de un programa.
  - **Analizadores y estimadores de tiempos de ejecución**
    - Proporcionar información sobre la ejecución de un programa (tiempo de ejecución, consumo de CPU, etc.)
  - **Comprobación de interfaces**
    - Analizar la consistencia y la compleción de los flujos de información y de control entre los módulos de un sistema.



## Verificación y Validación – Otras Técnicas

- Otras Técnicas de Verificación y Validación
  - **Análisis de requisitos**
    - Buscar errores sintácticos, inconsistencias lógicas o ambigüedades entre las entradas del sistema, sus salidas, procesos y datos.
  - **Análisis de trazabilidad de requisitos**
    - Verificar que cada requisito del sistema está incluido en algún elemento software.
    - Garantizar que las pruebas que se realizan sobre dicho software permiten comprobar que se satisfacen los requisitos.
  - **Monitores de software**
    - Supervisar la ejecución de un programa para localizar posibles áreas ineficientes. Al finalizar la ejecución, el monitor genera informes que describen la utilización de los recursos del programa.