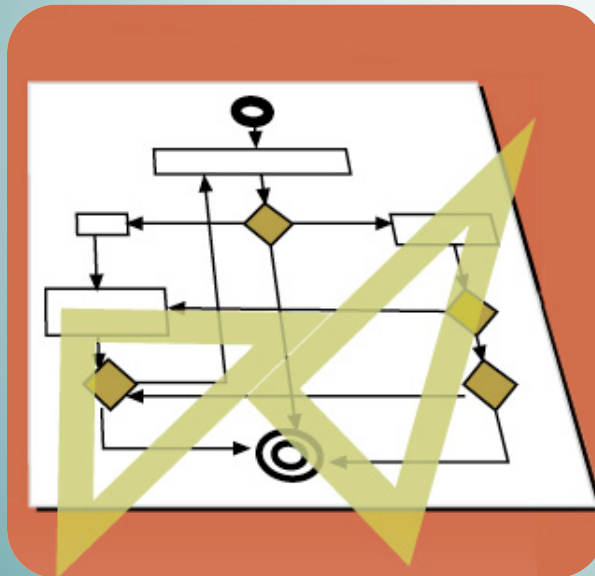


Ingeniería del Software II

Tema 02. Calidad del Software



Juan Hernández Marqués

DPTO. DE MATEMÁTICAS, ESTADÍSTICA Y
COMPUTACIÓN

juan.hernandez@unican.es

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 3.0](https://creativecommons.org/licenses/by-nc-sa/3.0/)



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
 - Historia de la Calidad
 - Concepto de Calidad
 - Perspectivas de la Calidad
 - Modelo de Calidad Total
 - Calidad de Sistemas de Información
 - Factores Calidad del Software
- Calidad de Producto
 - ISO 9126
 - Calidad de Datos
- Calidad de Proceso
 - ISO 90003
 - Evaluación y Mejora de Procesos
 - CMMI
 - ISO 15504 SPICE
 - PSP Y TSP
- Medición del Software
 - Proceso
 - GQM
 - Métricas
- Gestión de la Calidad del Sw
 - Aseguramiento
 - Verificación y Validación
 - Revisiones
 - Auditorias
- Apéndice A: Conceptos ISO 9000:2005
- Apéndice B: Otras Técnicas de Verificación y Validación
- Apéndice C: Lista de Defectos de las Inspecciones



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.



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 – Historia de la Calidad (1/2)

Etapas	Concepto	Finalidad
Artesanal	Hacer las cosas bien sin mirar coste o esfuerzo necesario para ello.	<ul style="list-style-type: none">• Satisfacer al Cliente.• Autosatisfacción del Artesano.• Crear un producto único.
Revolución Industrial	Hacer muchas cosas no importando que sean de calidad. (Producción = Calidad).	<ul style="list-style-type: none">• Satisfacer una gran demanda de bienes.• Obtener beneficios.
II Guerra Mundial	Asegurar la eficacia del armamento con la mayor y más rápida producción. No importa el coste. (Eficacia + Plazo = Calidad)	<ul style="list-style-type: none">• Garantizar la disponibilidad de armamento eficaz en cantidad y plazo.
Postguerra	(JAPÓN) Hacer las cosas bien a la primera	<ul style="list-style-type: none">• Minimizar costes mediante la Calidad• Satisfacer al cliente• Ser competitivo
	(RESTO) Producir cuanto más mejor	<ul style="list-style-type: none">• Satisfacer la gran demanda de bienes causada por la guerra



Introducción – Historia de la Calidad (2/2)

Etapa	Concepto	Finalidad
Control de Calidad	Inspección en producción para evitar bienes defectuosos.	<ul style="list-style-type: none">• Satisfacer las necesidades técnicas del producto.
Aseguramiento de la Calidad	Sistemas y Procedimientos de la organización para evitar que se produzcan bienes defectuosos.	<ul style="list-style-type: none">• Satisfacer al cliente.• Prevenir errores.• Reducir costes y ser competitivo.
Calidad Total	Administración empresarial centrada en la permanente satisfacción de las expectativas del cliente.	<ul style="list-style-type: none">• Satisfacer al cliente externo e interno.• Ser altamente competitivo.• Mejora Continua.

- 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 **Coste** y **Plazo**.
- Gran Importancia y atención a las **Expectativas del Cliente** (Investigación de Mercados, Especificaciones, ...).





Introducción – Concepto de Calidad

■ 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.*

■ ¿Qué tiene más calidad?





Introducción – Concepto de Calidad

- **¿Y cuáles eran mis requisitos para comprar un coche?**
 - No voy a hacer trayectos de más de 10 Km.
 - La ocupación máxima será de tres personas.
 - Quiero gastarme poco en seguro, consumo y mantenimiento.
- **Según esto, el elegido es....**



“La Calidad puede no ser lo que piensas”

P.B. Crosby



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?

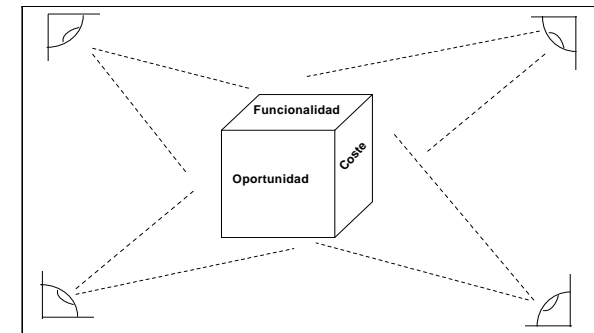
- **Definición Profesional:**

- Totalidad de las **características y aspectos de un producto o servicio** en los que se basa su aptitud para **satisfacer una necesidad** dada.
- Grado en el que un conjunto de características inherentes **cumple con los requisitos** (ISO 9000:2005).



Introducción – Concepto de Calidad

- **Calidad** es un concepto:
 - **Relativo**
 - La calidad está en los ojos del observador y es relativa a las personas, su edad y circunstancias, al espacio, tiempo, ...
 - **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**).

***“Calidad significa hacer lo correcto
cuando nadie está mirando”***

Henry Ford



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

- **Gestión de la Calidad** (*Quality Management*):
 - Actividades coordinadas para **dirigir y controlar una organización en lo relativo a la calidad**, aplicando 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., aunque también puede haber una gestión de calidad dentro de la gestión de cada proyecto.
- **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 porque se ha seguido un proceso de confección auditado.



Introducción – Concepto de Calidad

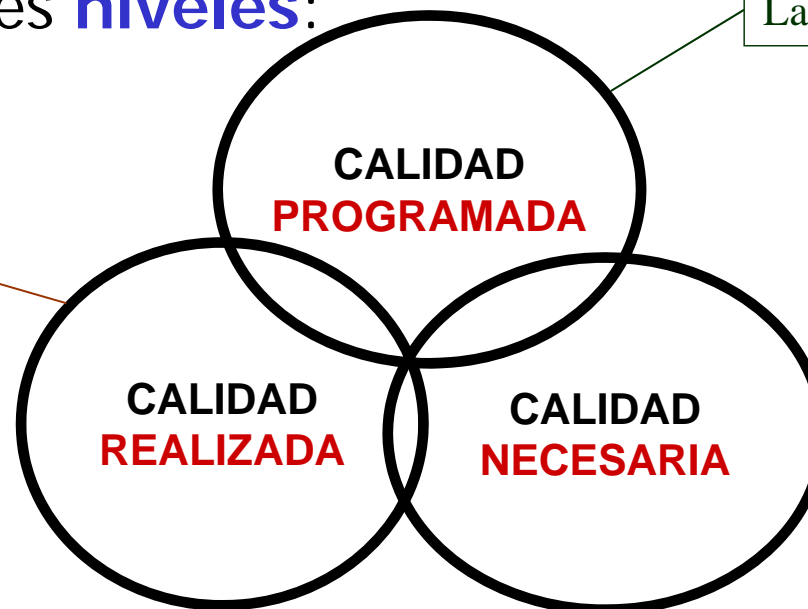
- **Control de la Calidad** (*Quality Control*):
 - Parte de la gestión de la calidad orientada al **cumplimiento de los requisitos de la calidad**.
 - Suele incluir técnicas y actividades de carácter operativo utilizadas para satisfacer los requisitos relativos a la calidad, centradas en dos objetivos 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)
- Y con diferentes **niveles**:

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

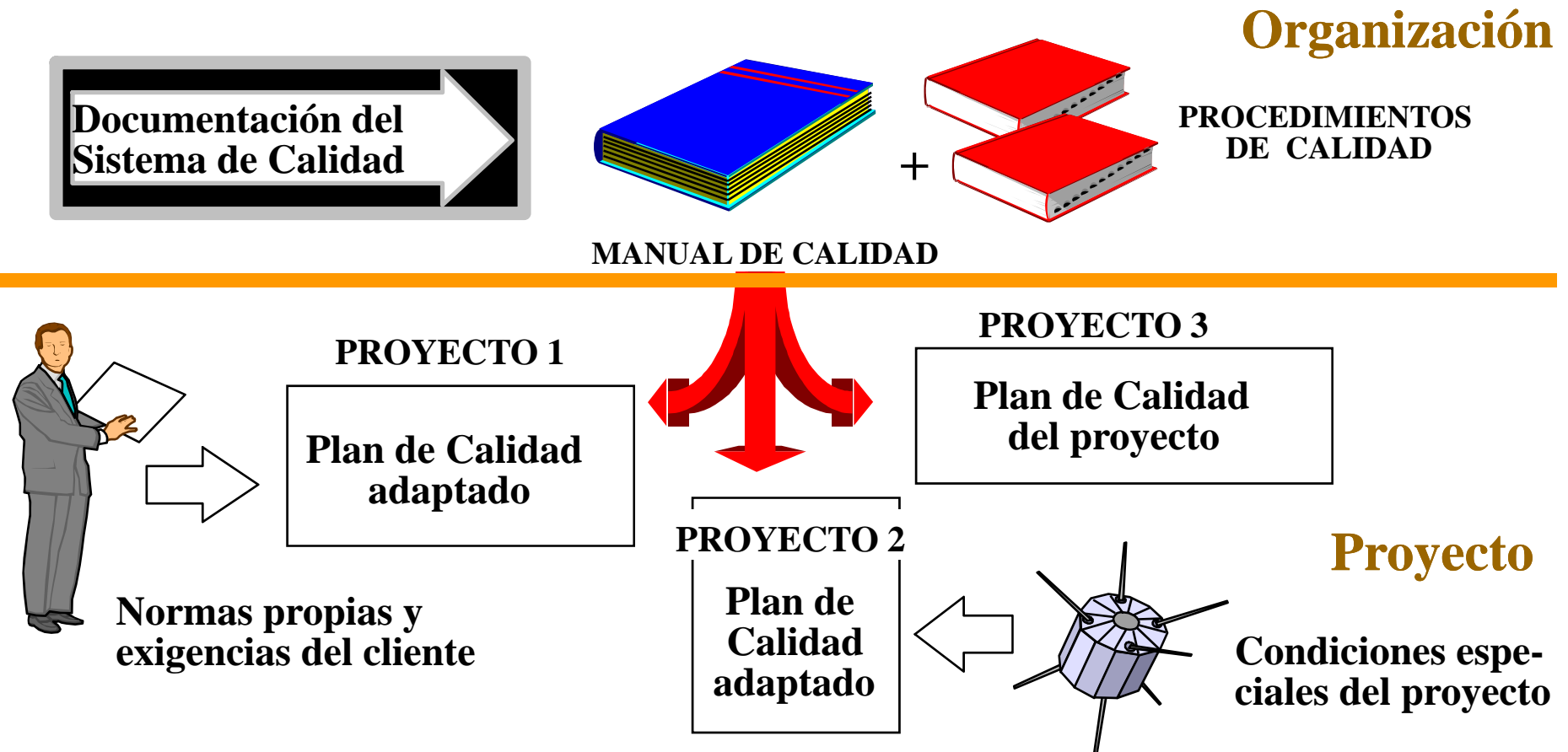
- La gestión de la calidad busca conseguir que estos tres círculos coincidan entre sí.
 - Todo lo que esté fuera de dicha coincidencia será motivo de derroche, de gasto superfluo o de insatisfacción.





Introducción – Perspectivas de la Calidad

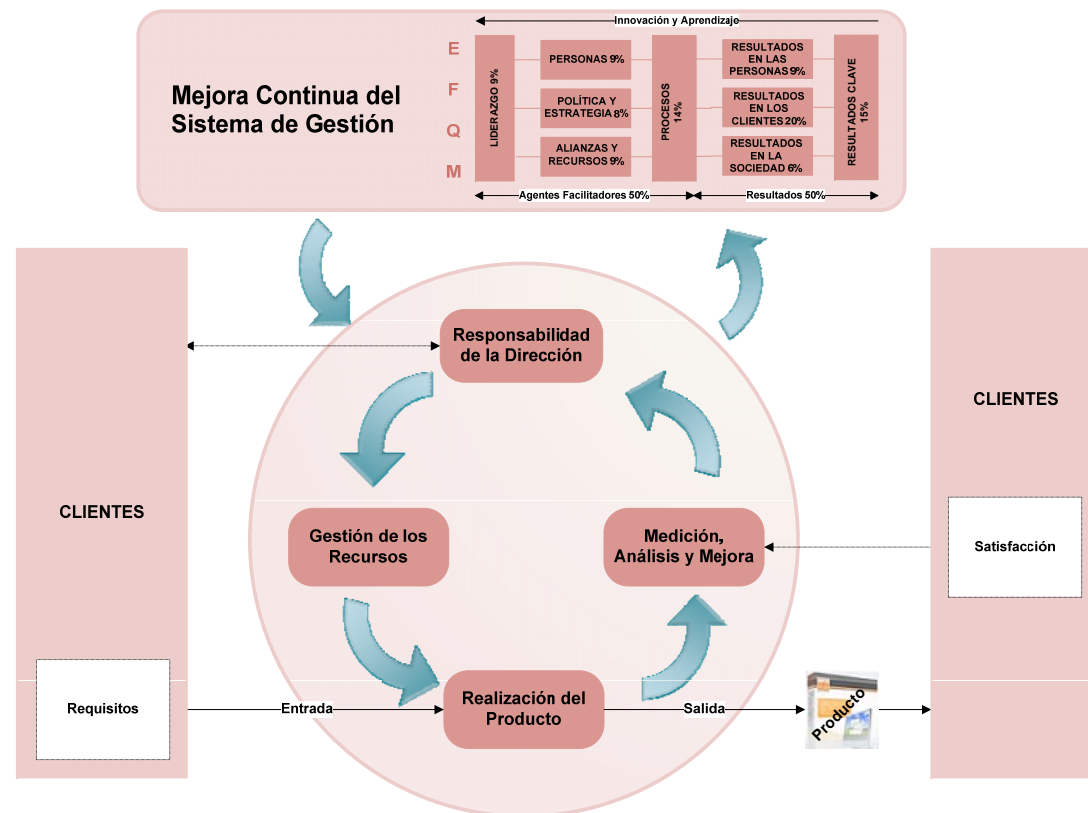
LOS DOS NIVELES DE LA CALIDAD Y LAS RELACIONES ENTRE AMBOS





Introducción – Modelo de Calidad Total

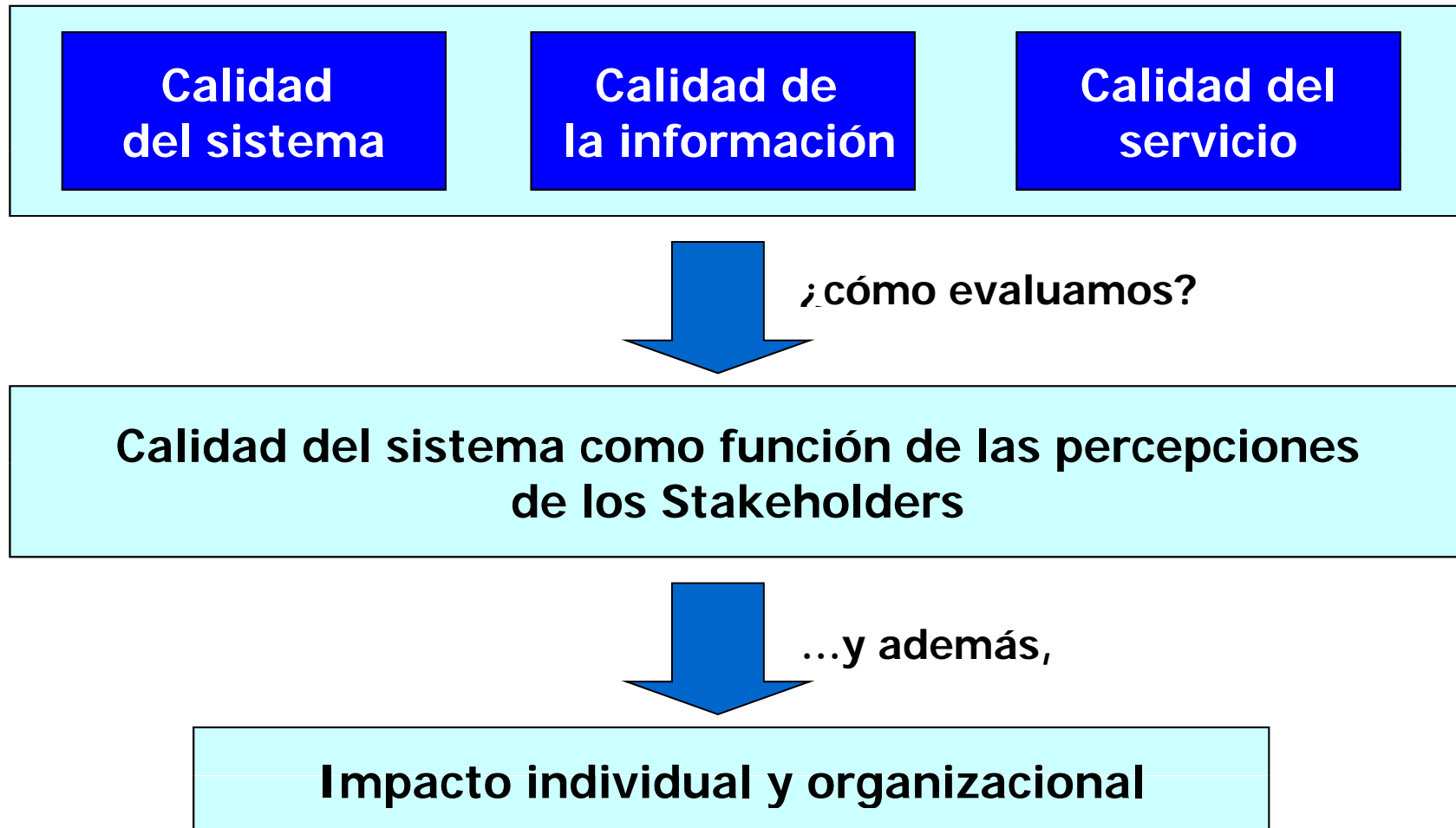
- La **norma ISO 9001:2008** es la principal referencial internacional sobre conceptos de calidad.
 - UNE-EN ISO 9000:2005 Sistemas de Gestión de la Calidad. Fundamentos y Vocabulario.*
- EFQM** modelo de Gestión Empresarial basado en la Mejora Continua.





Introducción - Calidad de Sistemas de Información

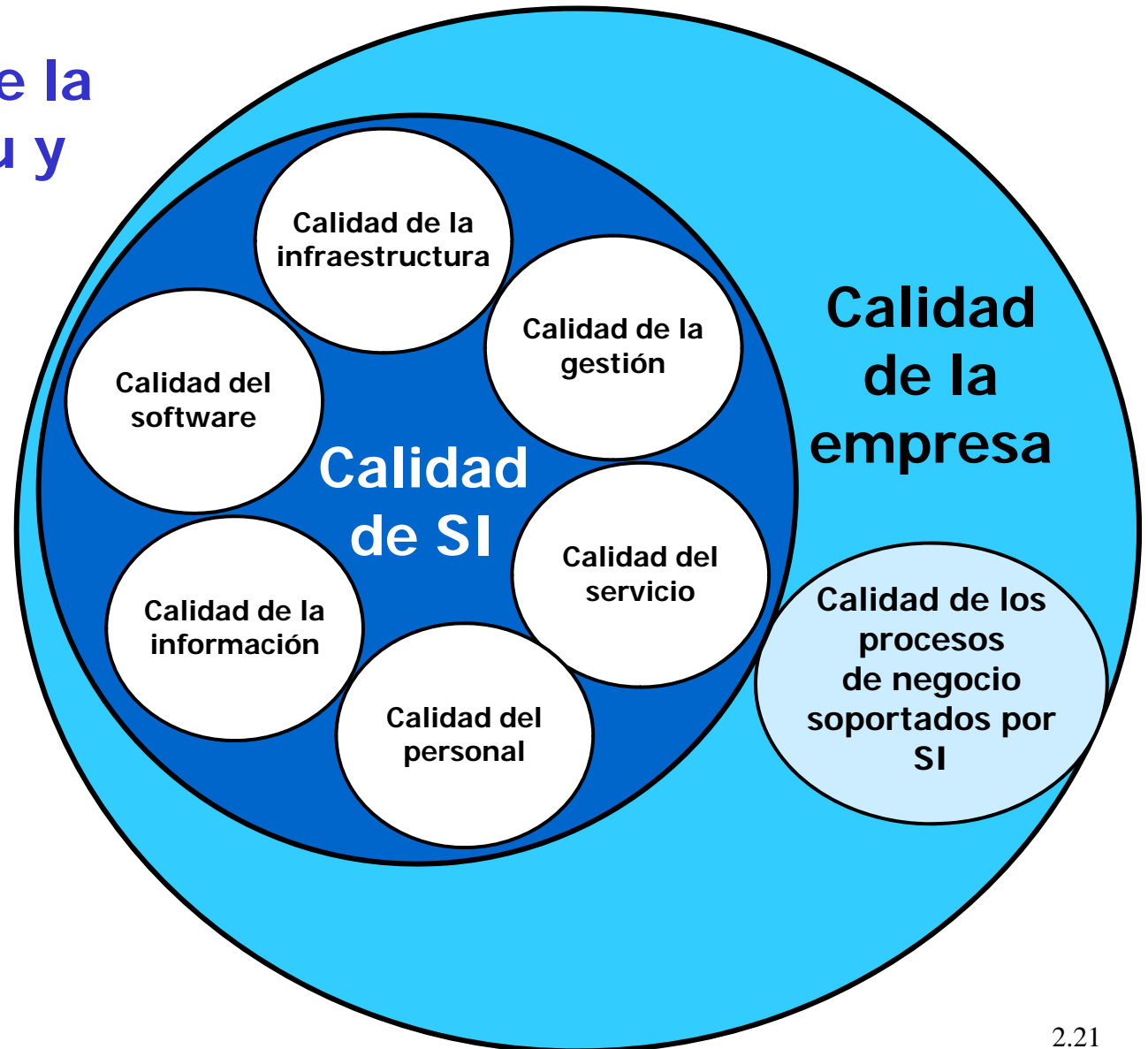
QUALIT. C. Wilkin y T. Castleman (2003)





Introducción - Calidad de Sistemas de Información

Visión holística de la calidad. Stylianou y Kumar (2000).

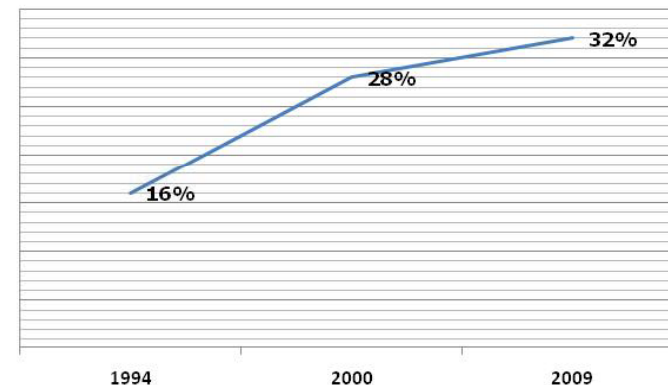
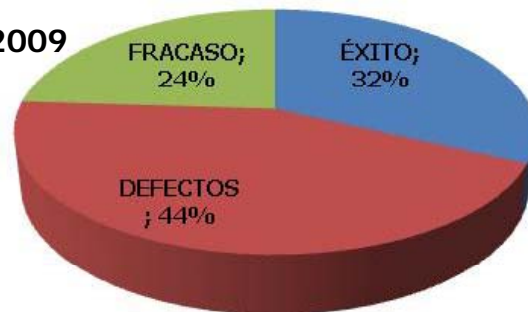




Introducción – Factores Calidad del Software

- “Crisis” en la 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).
- Informe “CHAOS” del Standish Group sobre éxito de los proyectos de software.

CHAOS Summary 2009



Éxito en Proyectos (CHAOS)



Introducción – Factores Calidad 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 y 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

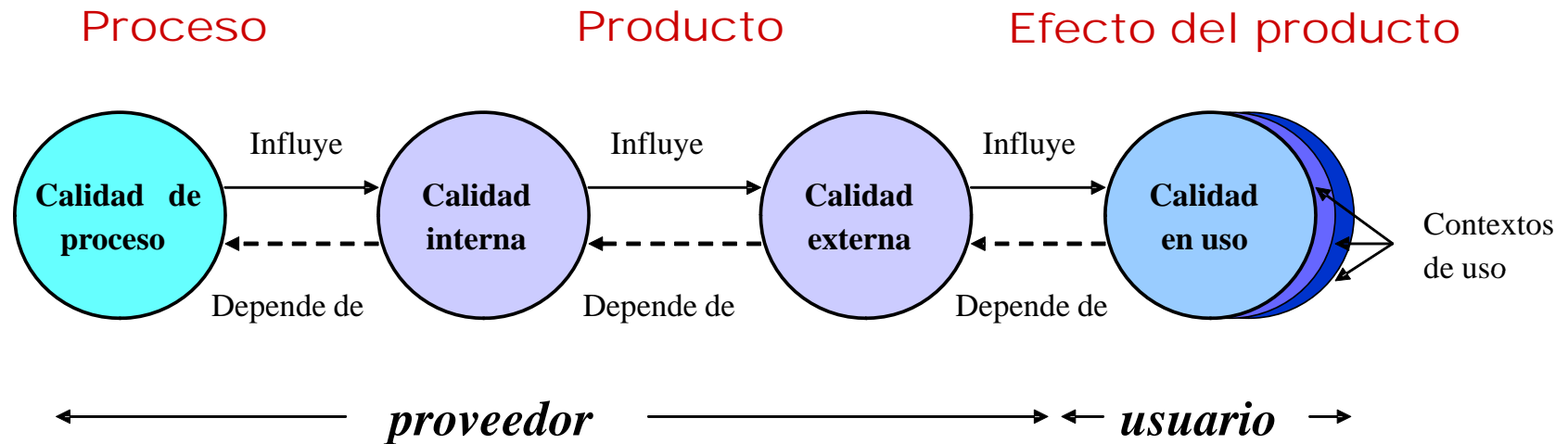
Calidad es el grado con el que un sistema, componente o proceso cumple los requisitos especificados y/o las necesidades o expectativas del cliente o usuario [IEEE Std.610-1991]

“Si McDonalds funcionara como una compañía de software, uno de cada cien Big Macs te envenenarían, y la respuesta sería ‘lo sentimos, aquí tiene un cupón para dos más’.

Mark Minasi



Introducción – Factores Calidad del Software



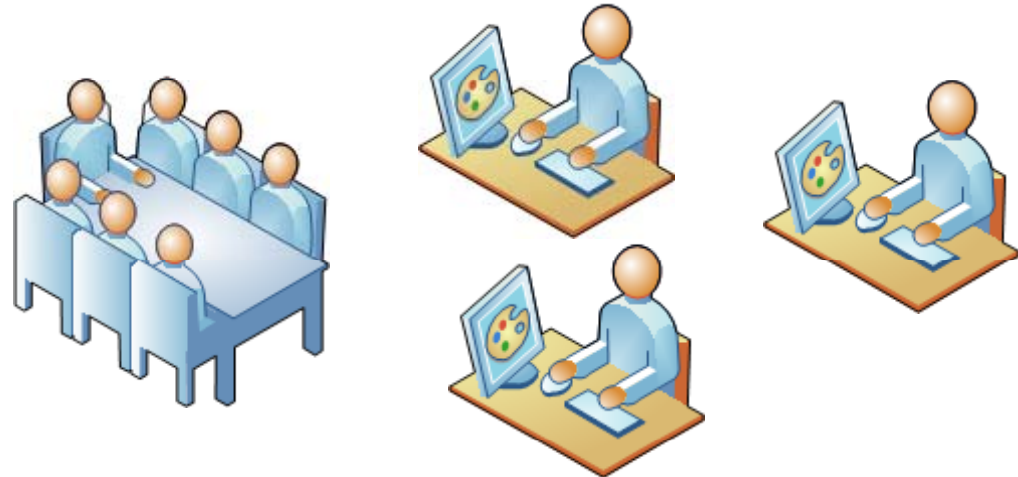
Calidad Interna	Calidad Externa	Calidad en Uso
medible a partir de las características intrínsecas	medible en el comportamiento del producto	durante la utilización efectiva por parte del usuario
Código fuente	Resultados de una prueba	Encuesta de satisfacción



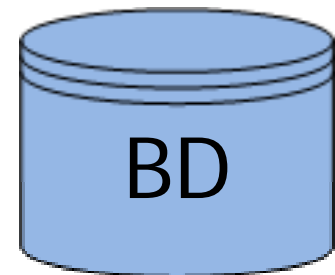
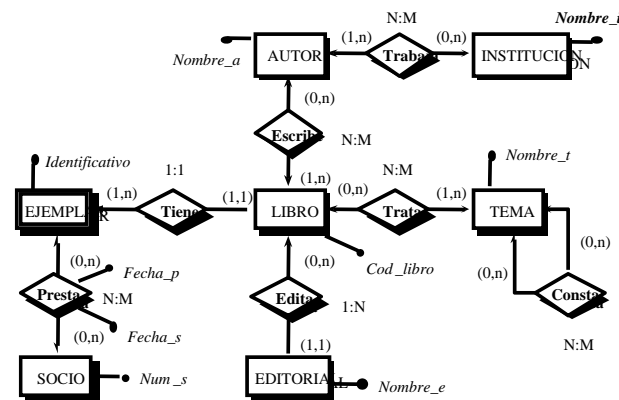
Calidad del Software – Factores Calidad del Software

Calidad del proceso

ISO 90003
*Evaluación y Mejora de
Procesos (CMMi, ISO 15504
SPICE, TSP, PSP)*



Calidad del producto



Software ISO 9126

Datos



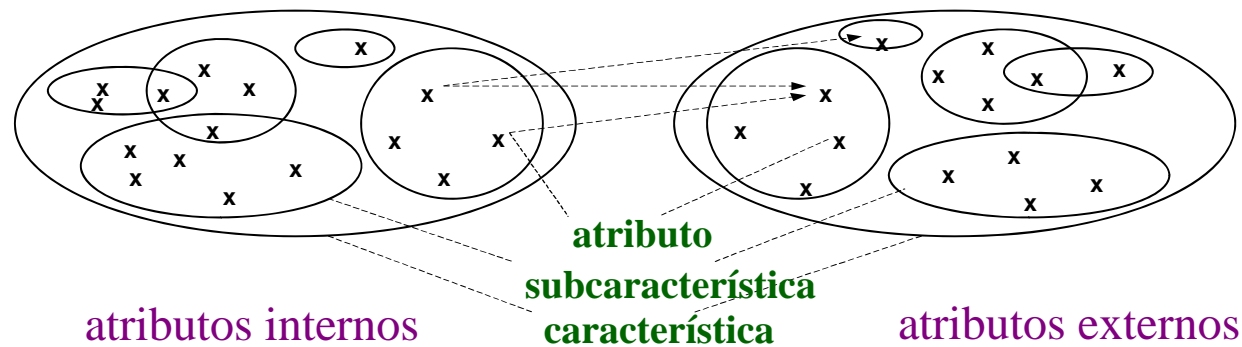
Calidad de Producto – ISO 9126

- **ISO/IEC 9126:** Tecnologías de la Información – Calidad de los Productos Software.
 - ISO/IEC 9126-1: Modelo de Calidad
 - ISO/IEC 9126-2: Métricas Externas
 - ISO/IEC 9126-3: Métricas Internas
 - ISO/IEC 9126-4: 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.



Calidad de Producto – ISO 9126

Identificación de Características, subcaracterísticas y atributos de calidad...



...para definir modelos de calidad

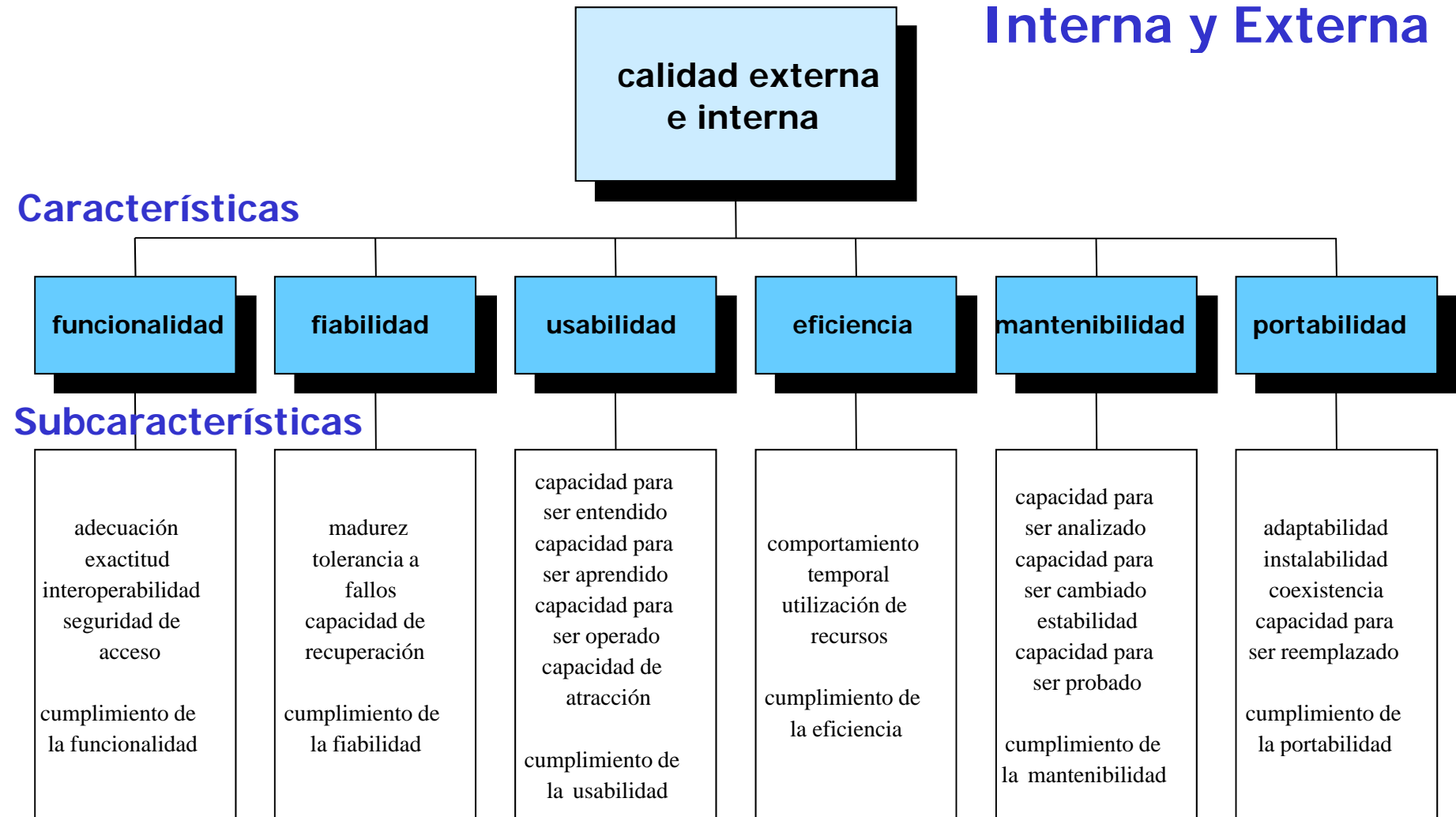
**Modelo de Calidad
Interna y Externa**

**Modelo de Calidad
En Uso**



Calidad de Producto – ISO 9126

Modelo de Calidad Interna y Externa





Calidad de Producto – ISO 9126 Características

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 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 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 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 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 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 Portabilidad

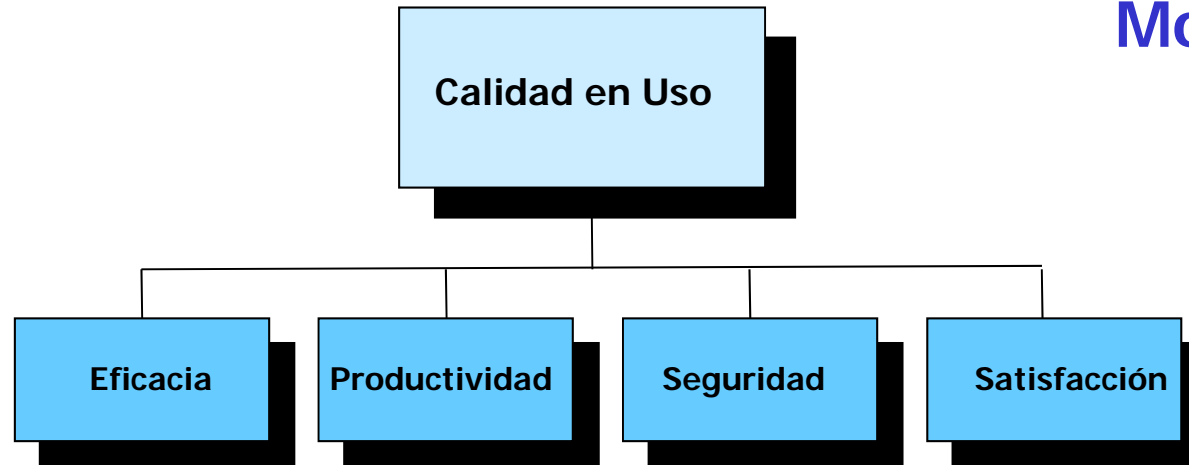
Capacidad del producto software para ... bajo condiciones especificadas.

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 – ISO 9126

Modelo de Calidad en Uso



Capacidad del producto software para ... bajo condiciones especificadas.

Eficacia	...para permitir a los usuarios alcanzar objetivos específicos con precisión y completamente...
Productividad	...para permitir a los usuarios emplear recursos apropiados con relación a la eficacia alcanzada...
Seguridad	...para alcanzar niveles aceptables de riesgo hacia la gente, negocio, software, propiedad o medio ambiente...
Satisfacción	...para satisfacer al usuario...



Calidad de Producto - Datos

- 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.
- Los datos se convierten en **fuentes de problemas**:
 - Datos no usados, inútiles e innecesarios y redundantes,
 - Barreras en la accesibilidad de los datos,
 - Dificultades en la utilización de los datos y de la información y gran cantidad de datos históricos caducados.
- 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 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 da a esos procesos.



Calidad de Producto - Datos

- Ejemplos de falta de calidad

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

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

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

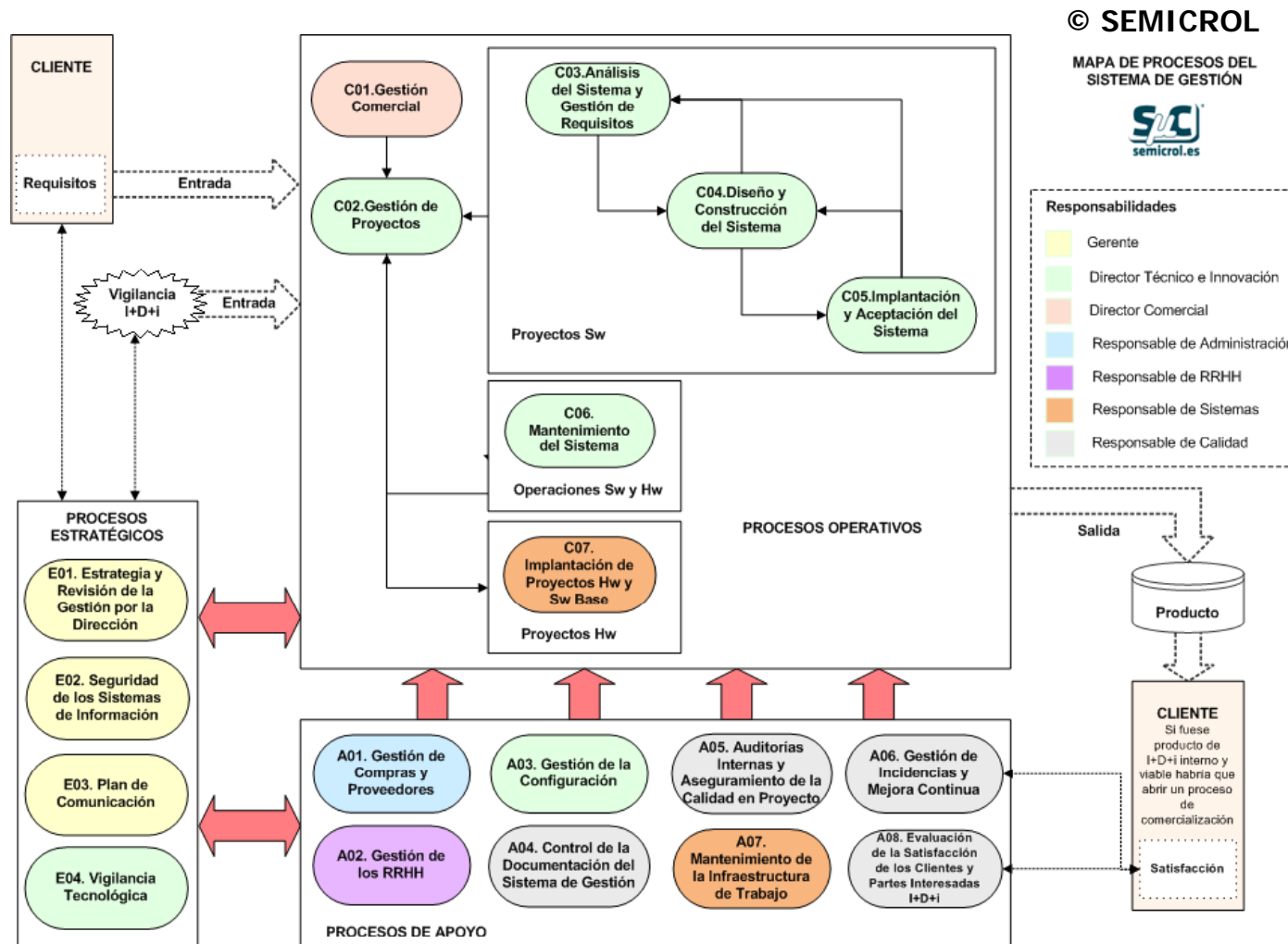


Calidad de Proceso – ISO 90003

- La norma **ISO 90003** proporciona, a las organizaciones, una guía para la **adaptación de la ISO 9001:2008** para la adquisición, suministro, desarrollo, instalación y mantenimiento de **SOFTWARE** y servicios de soporte. Identifica todos los aspectos que deberían ser tratados y es independiente de la tecnología, modelos de ciclo de vida, procesos de desarrollo y estructuras organizacionales.
- 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.



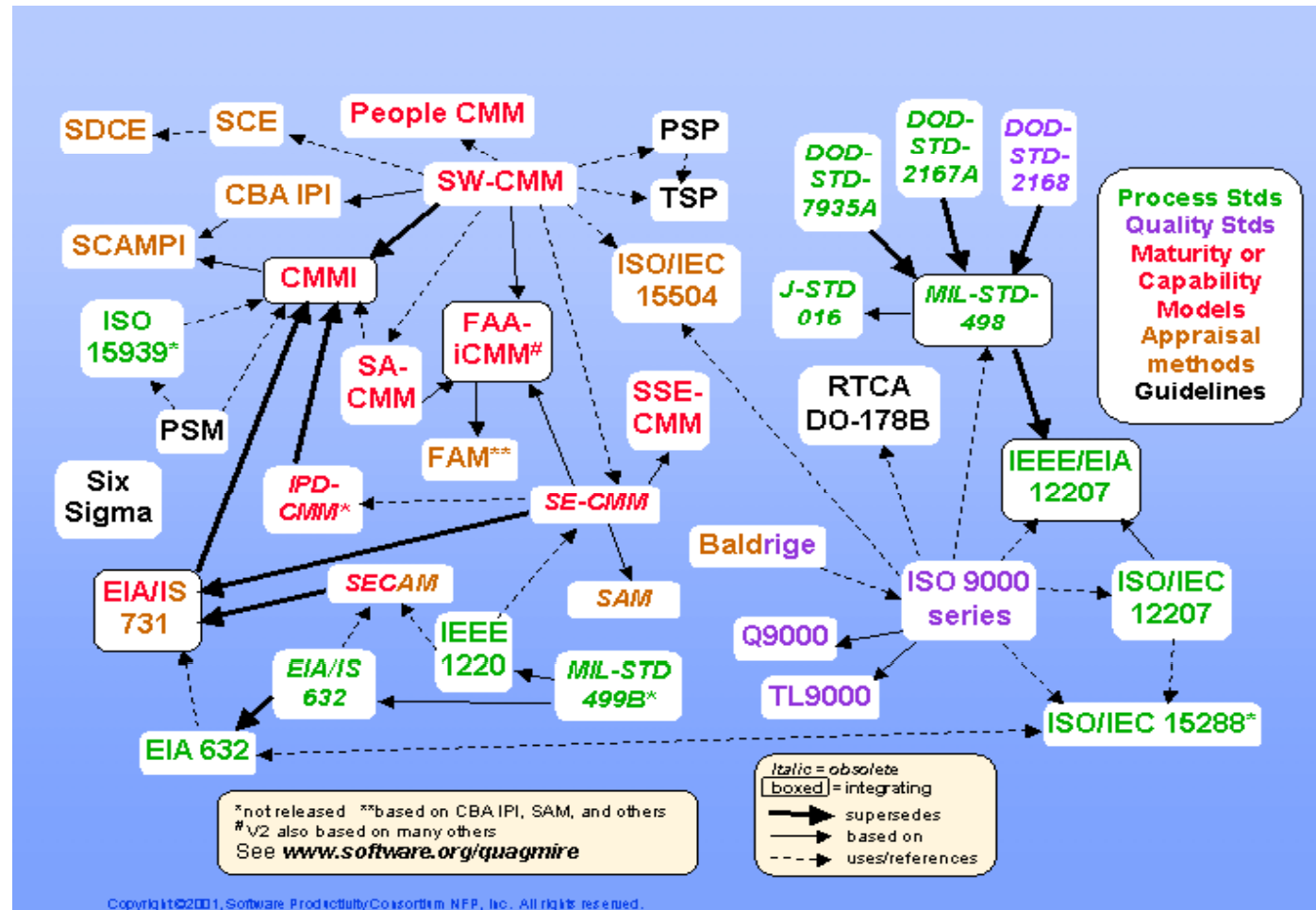
Calidad de Proceso – Ejemplo Mapa Procesos





Calidad de Proceso - 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 - CMMI

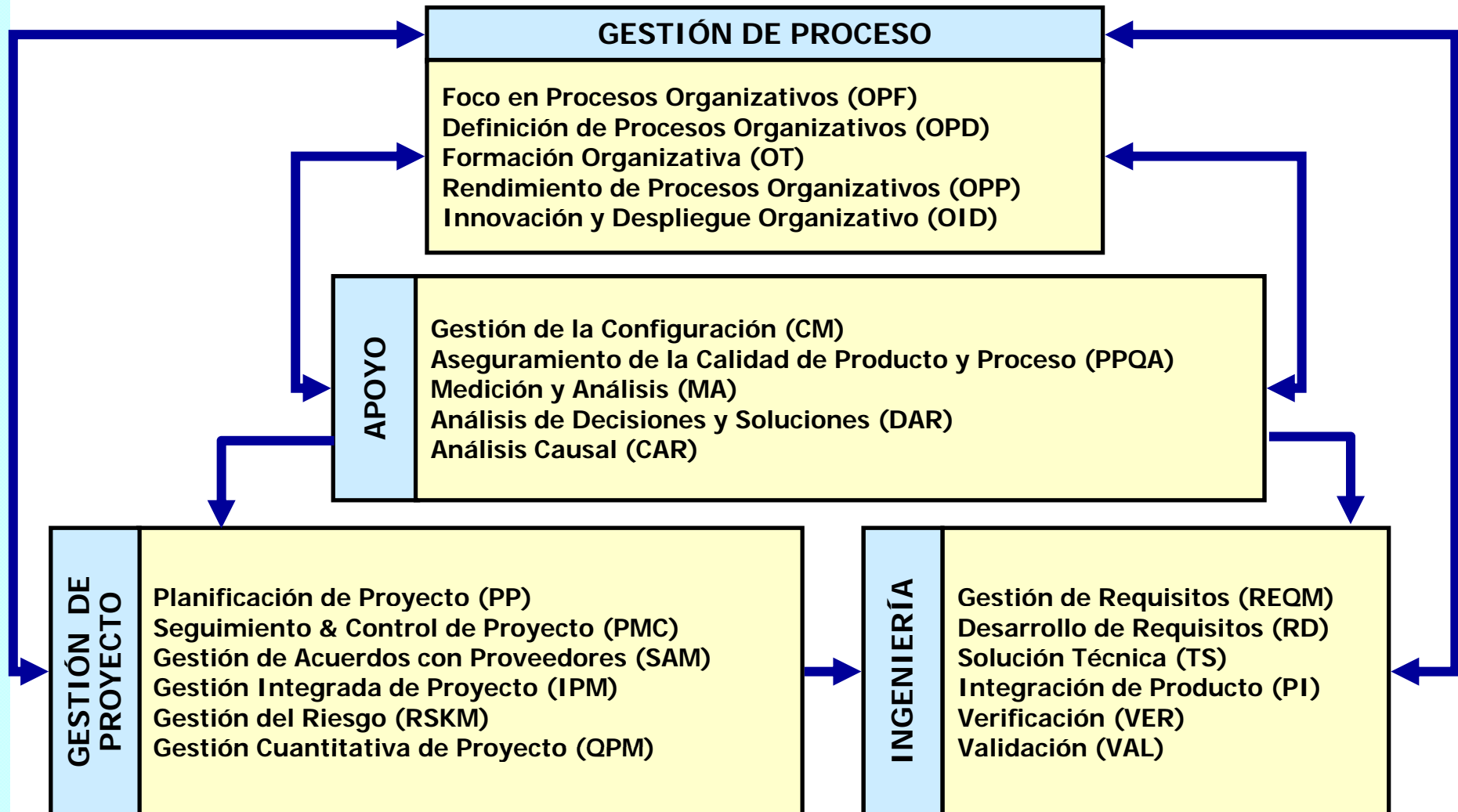
- CMMI (**Capability Maturity Model Integrated**) proporciona a las organizaciones de software el **modelo de referencia** necesario como soporte para el **control** de sus **procesos de desarrollo y mantenimiento** y para facilitar su evolución hacia una cultura de la Ingeniería del Software y de excelencia en la gestión.
 - Desarrollado por el SEI (*Software Engineering Institute*) de la Universidad de Carnegie Mellon, USA. Es la evolución del anterior CMM.
 - 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.
 - Existen, en la v1.2, tres **modelos**,

Modelo	Descripción	Extensión
CMMI-DEV SW Ago-06	CMMI for Development	CMMI-DEV IPPD
CMMI-ACQ Nov-07	CMMI for Acquisition	
CMMI-SEV Feb-09	CMMI for Services	



Evaluación y Mejora de Procesos - CMMI

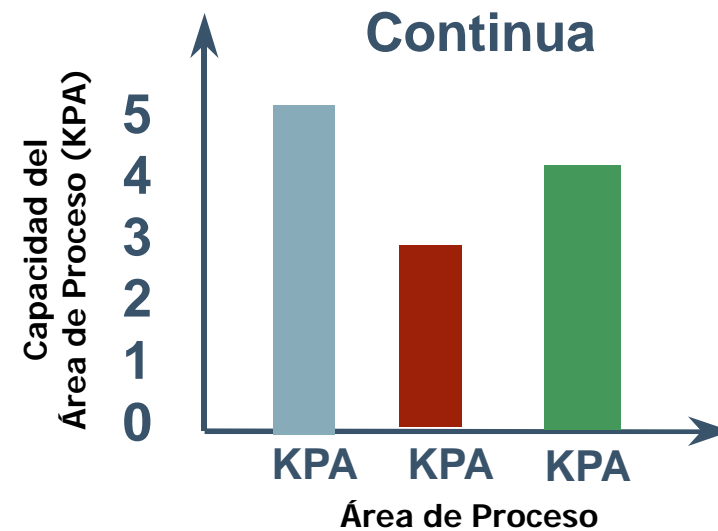
- Áreas de Proceso Claves (KPAs)





Evaluación y Mejora de Procesos – CMMI

- Hay dos **representaciones** de **CMMI-DEV SW**,



- 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.
- Muestra la representación del nivel de **capacidad de la organización para cada una de las áreas de proceso**.



Evaluación y Mejora de Procesos - CMMI

• Niveles de Madurez.

Es el típico proyecto en el que se da la siguiente situación:

- ¿Cómo va el proyecto?
- Bien, bien.

Dos semanas después...

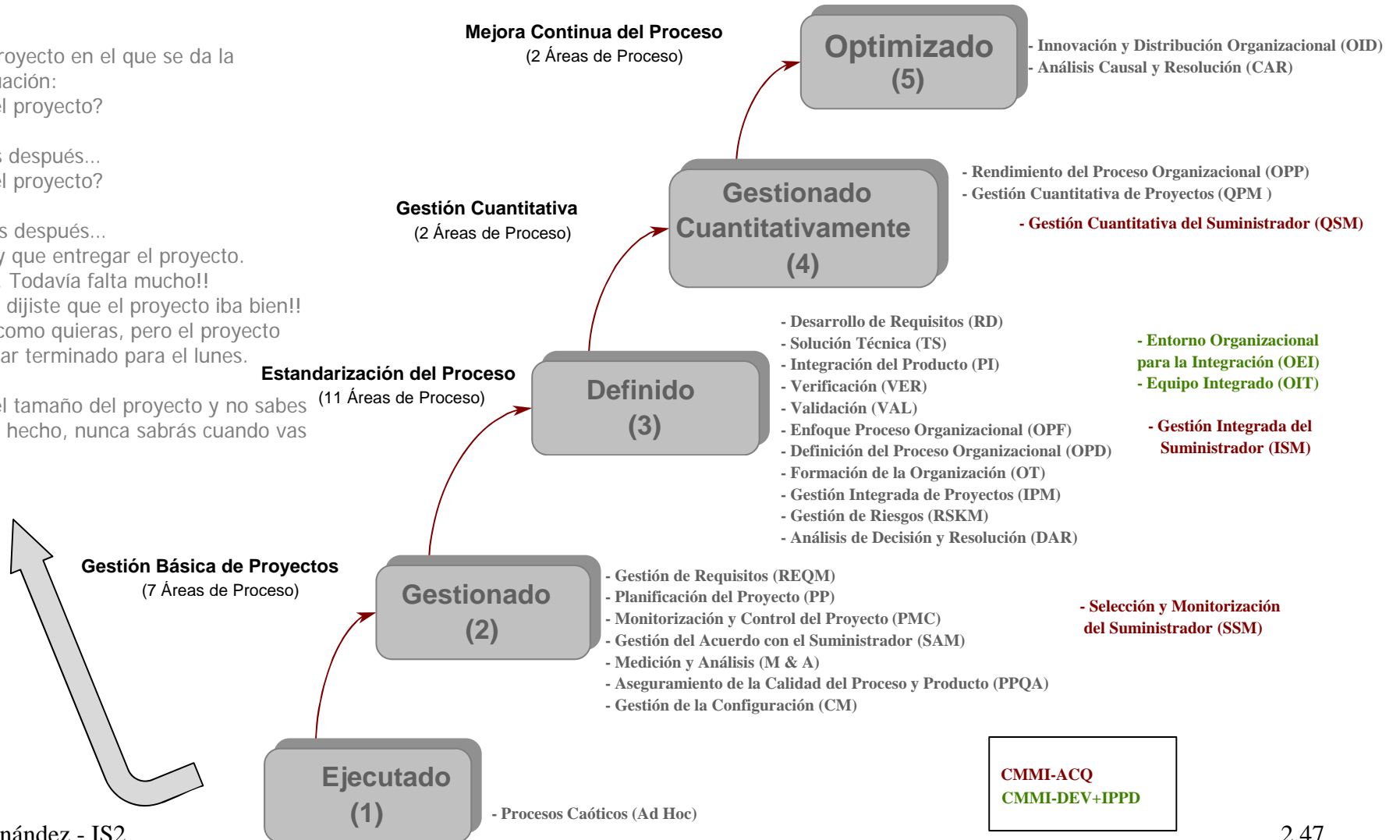
- ¿Cómo va el proyecto?
- Bien, bien.

Tres semanas después...

- El lunes hay que entregar el proyecto.
- El lunes !!?. Todavía falta mucho!!
- ¿Cómo? Me dijiste que el proyecto iba bien!!

Arréglatelas como quieras, pero el proyecto tiene que estar terminado para el lunes.

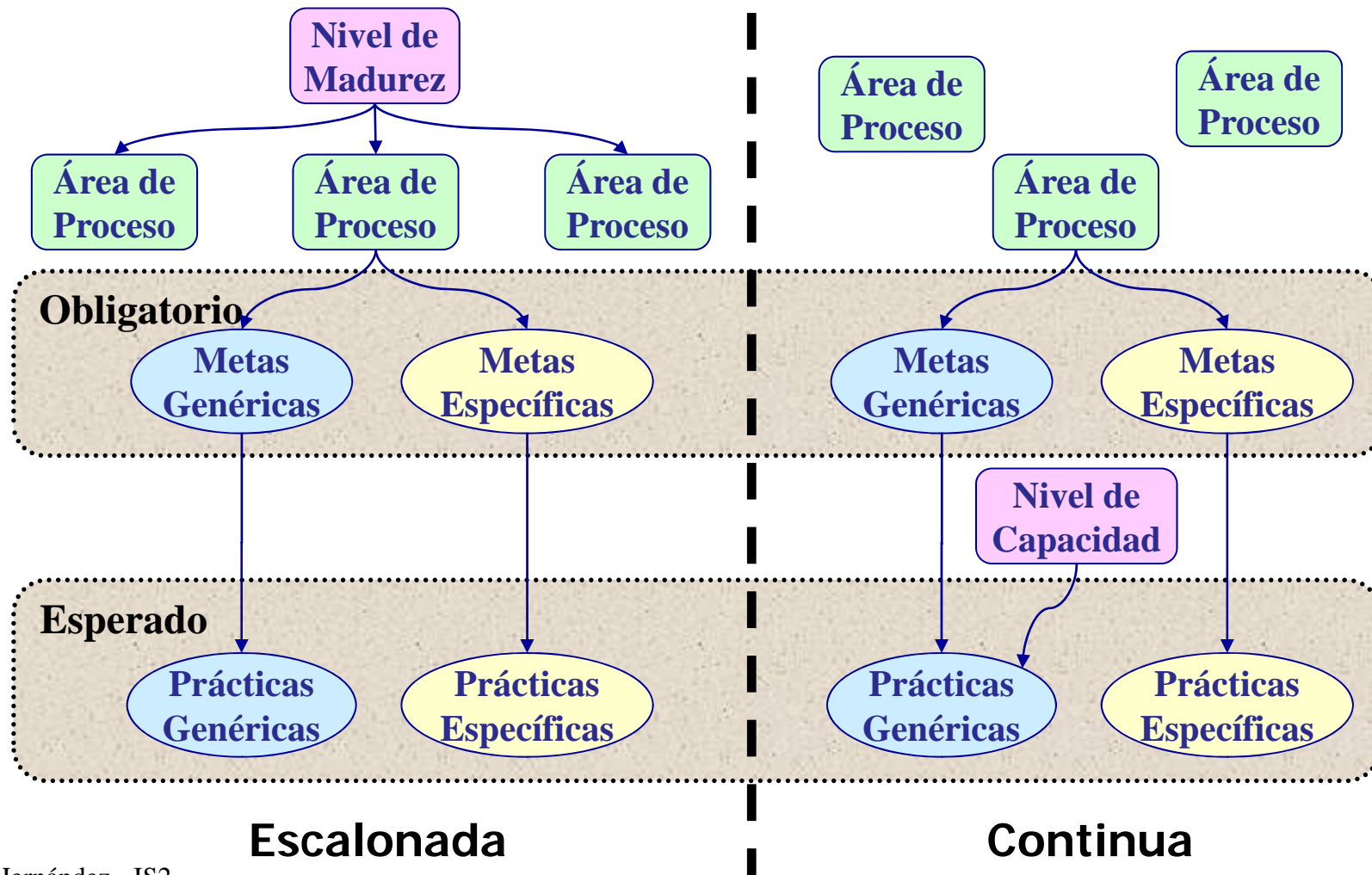
Si no sabes el tamaño del proyecto y no sabes cuanto llevas hecho, nunca sabrás cuando vas a terminar.





Evaluación y Mejora de Procesos - CMMI

- Estructura del modelo





Evaluación y Mejora de Procesos - CMMI

- **Ejemplo de KPA, Project Planning (PP)**
- **SG1: Establish Estimates**
 - SP1.1 Estimate the Scope of the Project
 - SP1.2 Establish Estimates of Work Product and Task Attributes
 - SP1.3 Define Project Lifecycle
 - SP1.4 Determine Estimates of Effort and Cost
- **SG2: Develop a Project Plan**
 - SP2.1 Establish the Budget and Schedule
 - SP2.2 Identify Project Risks
 - SP2.3 Plan for Data Management
 - SP2.4 Plan for Project Resources
 - SP2.5 Plan for Needed Knowledge and Skills
 - SP2.6 Plan Stakeholder Involvement
 - SP2.7 Establish the Project Plan
- **SG3: Obtain Commitment to the Plan**
 - SP3.1 Review Plans That Affect the Project
 - SP3.2 Reconcile Work and Resource Levels
 - SP3.3 Obtain Plan Commitment



Evaluación y Mejora de Procesos - CMMI

• Metas y Prácticas Genéricas

- GG1: Conseguir las metas específicas
 - **GP1: Realizar las prácticas específicas**
- GG2: Institucionalizar un Proceso **Gestionado**
 - **GP2.1: Establecer una política organizativa**
 - **GP2.2: Planificar el proceso**
 - **GP2.3: Suministrar recursos para la realización del proceso**
 - **GP2.4: Asignar responsabilidades para realizar el proceso**
 - **GP2.5: Entrenar a las personas que realizan el proceso**
 - **GP2.6: Gestionar la configuración de los elementos del proceso**
 - **GP2.7: Identificar e involucrar a los agentes relevantes del proceso**
 - **GP2.8: Seguir y controlar la realización del proceso**
 - **GP2.9: Evaluar objetivamente el cumplimiento del proceso**
 - **GP2.10: Revisar el estado con la Dirección**
- GG3: Institucionalizar un Proceso **Definido**
 - **GP3.1: Establecer un proceso definido**
 - **GP3.2: Recopilar información sobre la mejora del proceso**
- GG4: Institucionalizar un Proceso **Gestionado Cuantitativamente**
 - **GP4.1: Establecer objetivos cuantitativos para el proceso**
 - **GP4.2: Estabilizar el rendimiento de los subprocesos**
- GG5: Institucionalizar Proceso **Optimizado**
 - **GP5.1: Asegurar la mejora continua del proceso**
 - **GP5.2: Corregir la causa de los problemas**



Evaluación y Mejora de Procesos - CMMI

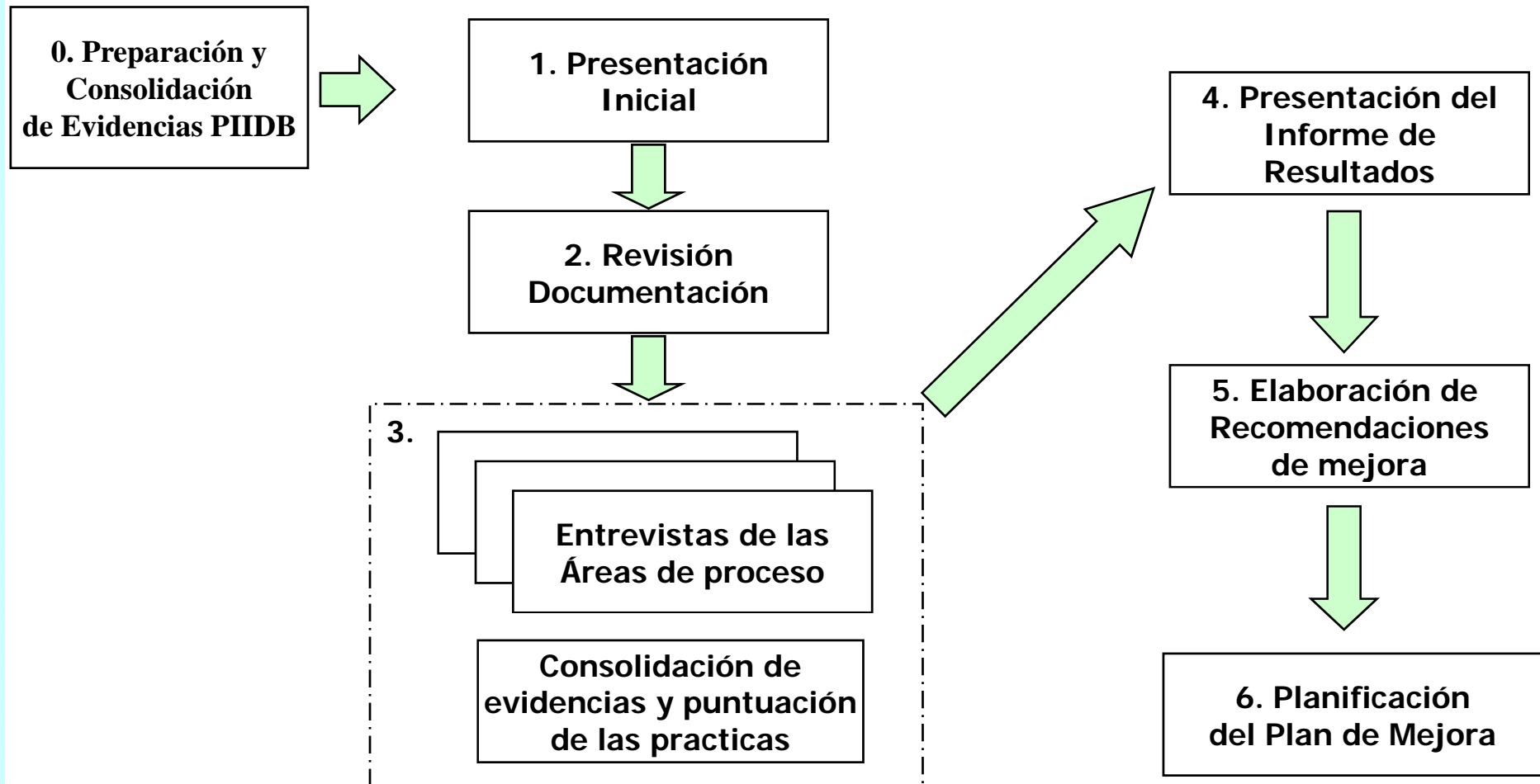
- **Evaluación (Appraisal)**

- Las organizaciones pueden querer evaluar su nivel de madurez (organizacional) o su nivel de capacidad (de procesos determinados) por varias razones:
 - Comparar con las mejores prácticas CMMI y determinar qué mejoras se pueden hacer.
 - Informar a los clientes externos y proveedores acerca de cómo funciona la organización y lleva a cabo sus procesos.
 - Para cumplir los requisitos contractuales de uno o más clientes.
- **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 del proceso de desarrollo, y
 - Determinar niveles de capacidad y madurez.



Evaluación y Mejora de Procesos - CMMI

- Evaluación (Appraisal)**





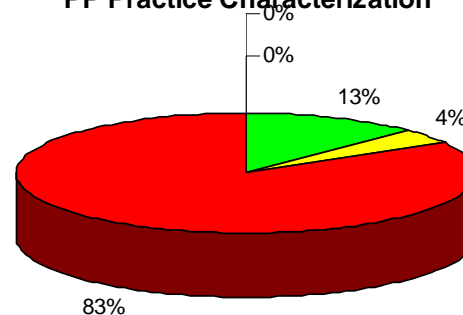
Evaluación y Mejora de Procesos - CMMI

Evaluación (Appraisal). Ejemplo Resultados PP.

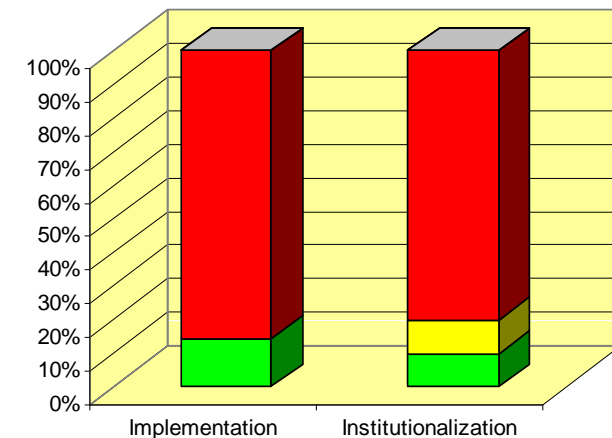
Metas	
●	SG1: Realizar estimaciones
●	SG2: Desarrollar un plan de proyecto
●	SG3: Obtener el compromiso con el plan
●	GG2: Institucionalizar un Proceso Gestionado

PA	Goals	Practices									
PP	SG1	SP1.1	SP1.2	SP1.3	SP1.4						
		R	R	G	R						
	SG2	SP2.1	SP2.2	SP2.3	SP2.4	SP2.5	SP2.6	SP2.7			
		R	R	G	R	R	R	R			
	SG3	SP3.1	SP3.2	SP3.3							
		R	R	R							
	GG2	GP2.1	GP2.2	GP2.3	GP2.4	GP2.5	GP2.6	GP2.7	GP2.8	GP2.9	GP2.10
		R	Y	R	R	R	G	R	R	R	R

PP Practice Characterization



PP Level of Implementation and Institutionalization





Evaluación y Mejora de Procesos - CMMI

- Evaluación (Appraisal). Ejemplo Resultados Nivel 2.**

Áreas de Proceso – Perfiles de Metas				
Área de Proceso	SG1	SG2	SG3	GG2
Gestión de Requisitos				
Planificación del Proyecto				
Seguimiento y Control del Proyecto				
Gestión de Acuerdos con Proveedores				
Medición y Análisis				
Aseguramiento de la Calidad				
Gestión de la Configuración				

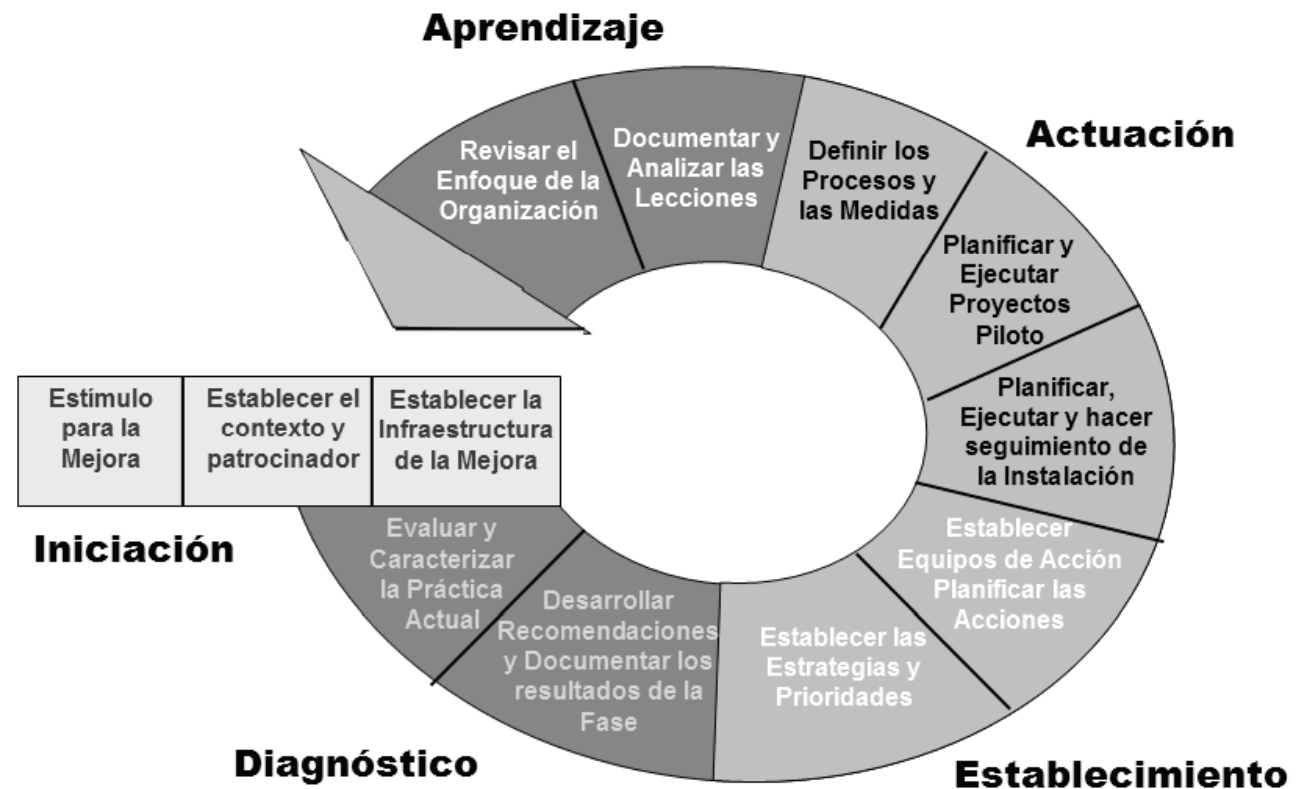
	5 de 22 → 23 %
	7 de 22 → 32 %
	10 de 22 → 45 %



Evaluación y Mejora de Procesos - CMMI

- **Mejora (Improvement)**

- **IDEAL** define un marco de ciclo de vida para la mejora de procesos.



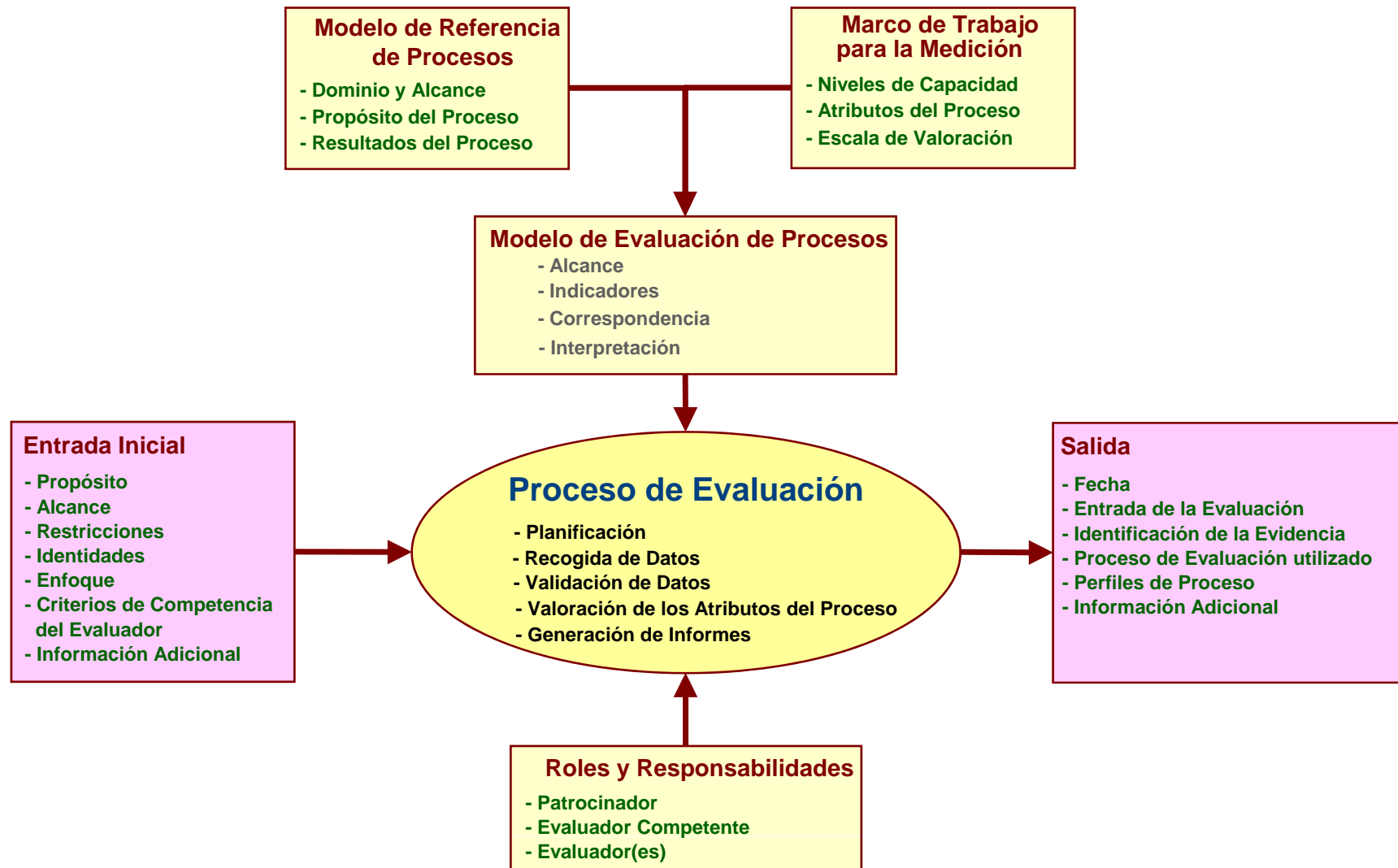


Evaluación y Mejora de Procesos – ISO 15504 SPICE

- 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).



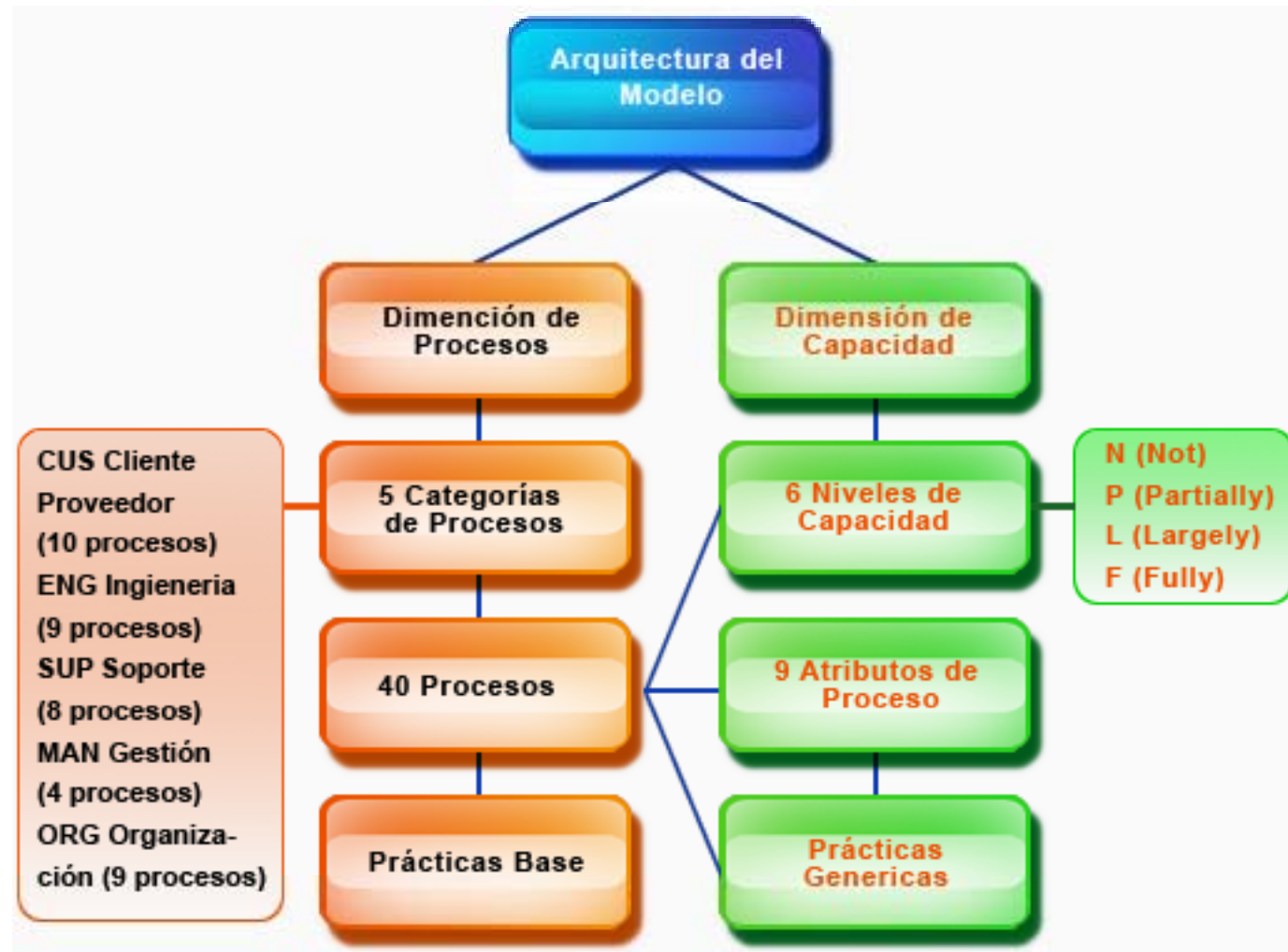
Evaluación y Mejora de Procesos – ISO 15504 SPICE





Evaluación y Mejora de Procesos – ISO 15504 SPICE

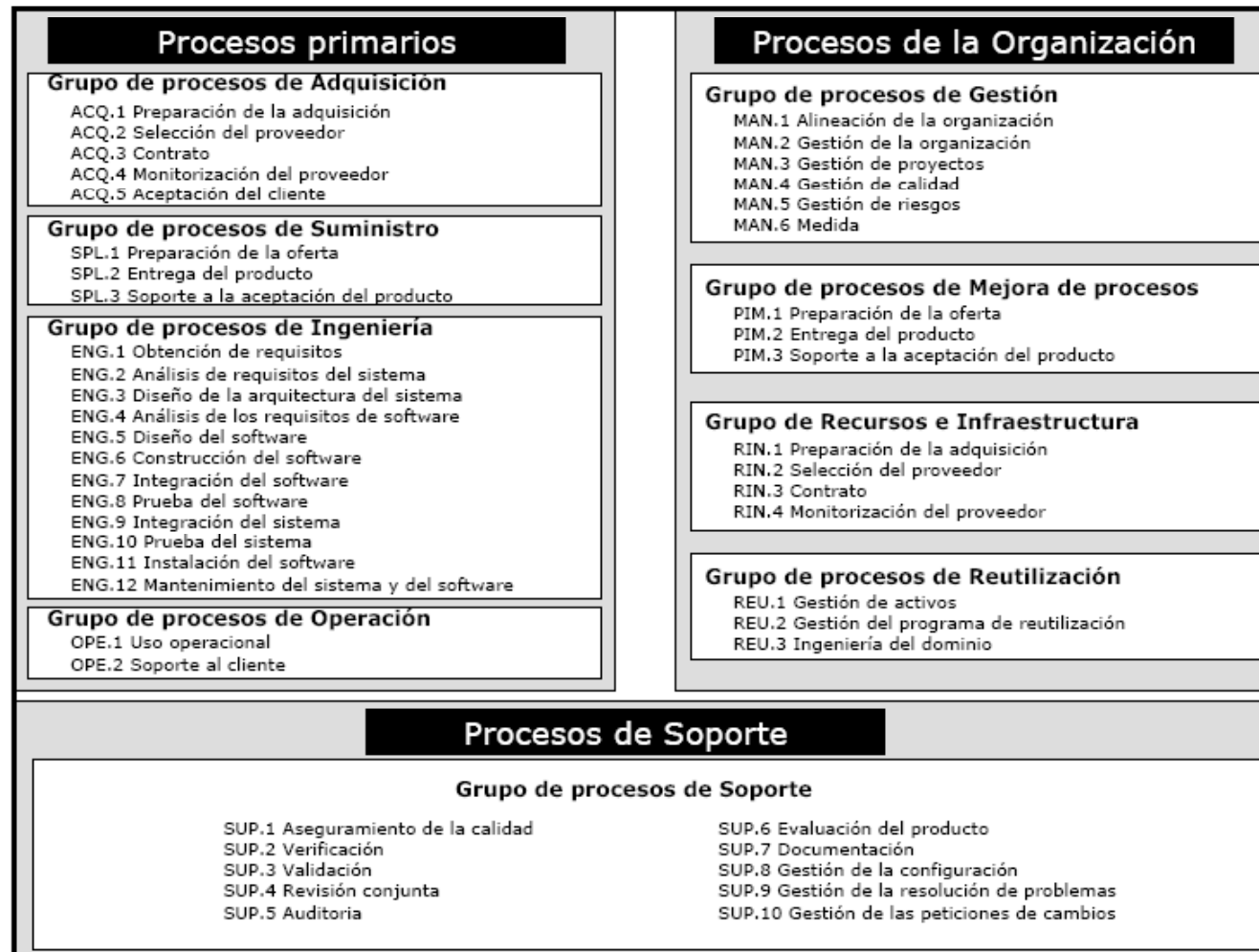
2 Dimensiones: **Proceso** y **Capacidad**





Evaluación y Mejora de Procesos – ISO 15504 SPICE

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

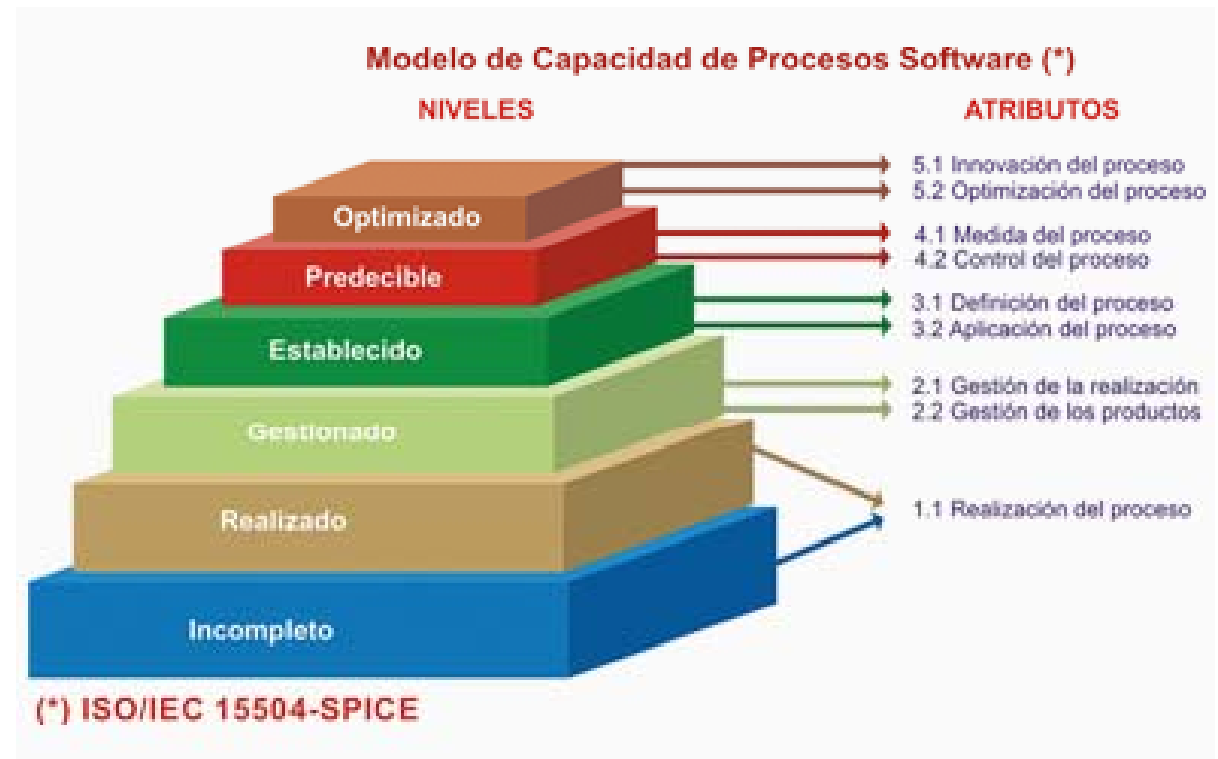




Evaluación y Mejora de Procesos – ISO 15504 SPICE

Modelo de Referencia de Capacidad

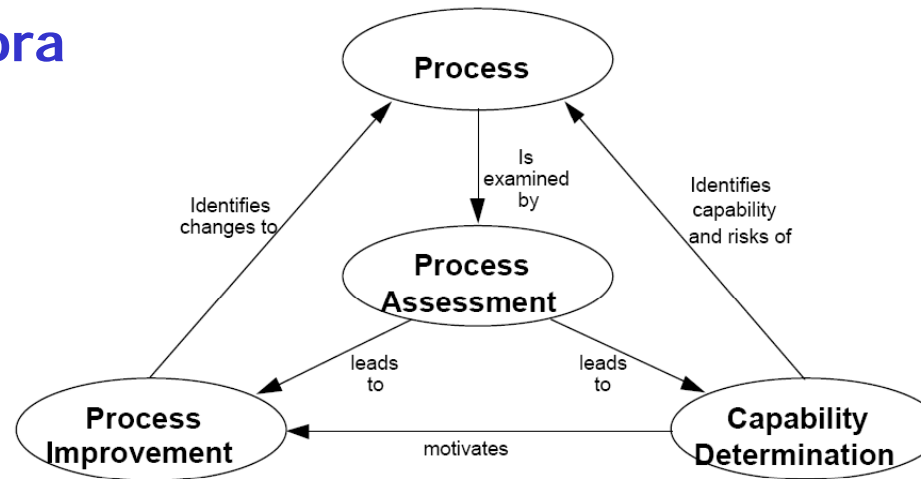
Process Attribute ID	Capability Levels and Process Attributes
	Level 0 : Incomplete process
	Level 1 : Performed process
PA 1.1	Process performance
	Level 2 : Managed process
PA 2.1	Performance management
PA 2.2	Work product management
	Level 3 : Established process
PA 3.1	Process definition
PA 3.2	Process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





Evaluación y Mejora de Procesos – ISO 15504 SPICE

Contextos de Mejora y Capacidad

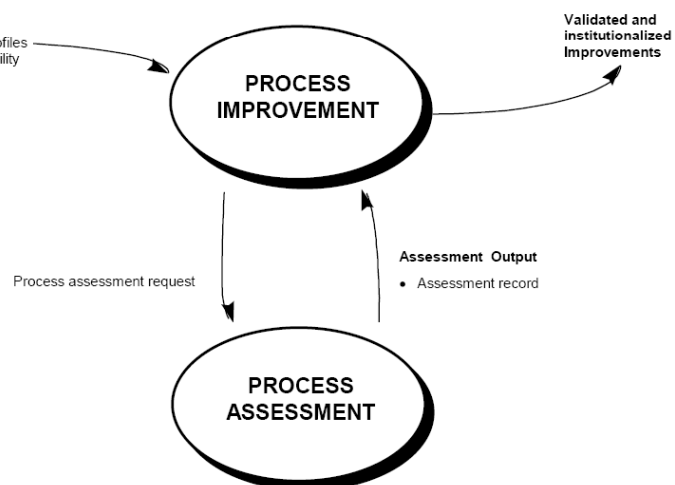


Software process improvement request

Organization's needs and business goals

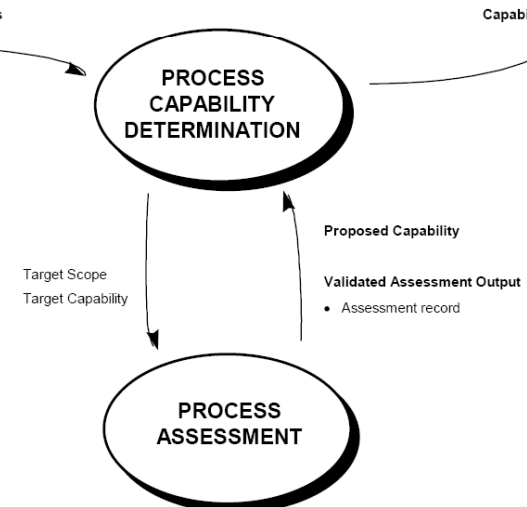
Target capability profiles from process capability determination

Industry norms and benchmarks



Specified Requirements

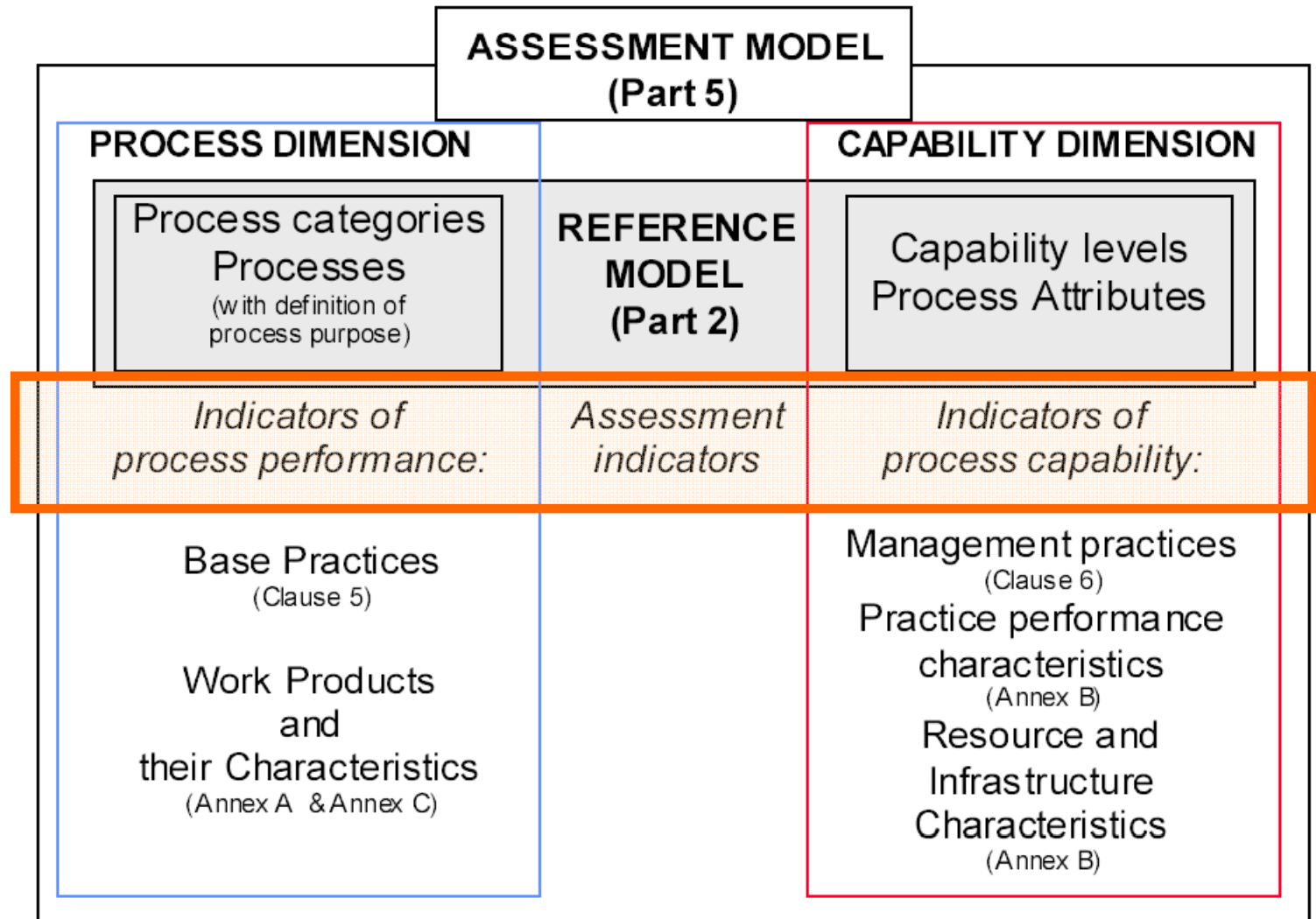
Capability Report





Evaluación y Mejora de Procesos – ISO 15504 SPICE

Modelo de Evaluación





Evaluación y Mejora de Procesos – ISO 15504 SPICE

Resultados de la Evaluación (Assessed Capability)

Process	Process Attributes								
	Performed			Managed		Established		Predictable	
	PA1			PA2.1		PA2.2		PA3.1	
	PA3.2			PA4.1		PA4.2		PA5.1	
Requirements elicitation									
Customer support									
Software design									
Software construction									
Software testing									

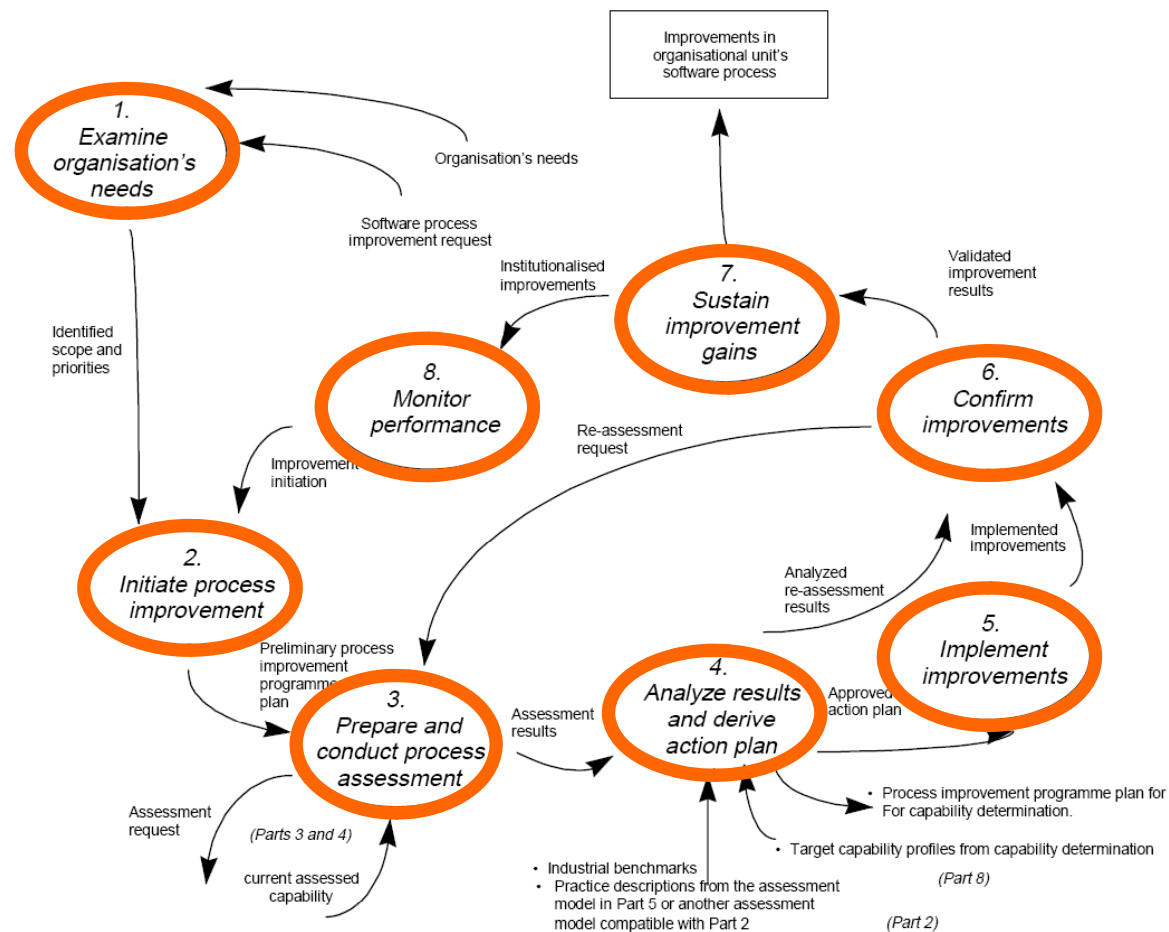
Key (as defined in Part 2)

	Fully Achieved		Largely Achieved
	Partially Achieved		Not Achieved



Evaluación y Mejora de Procesos – ISO 15504 SPICE

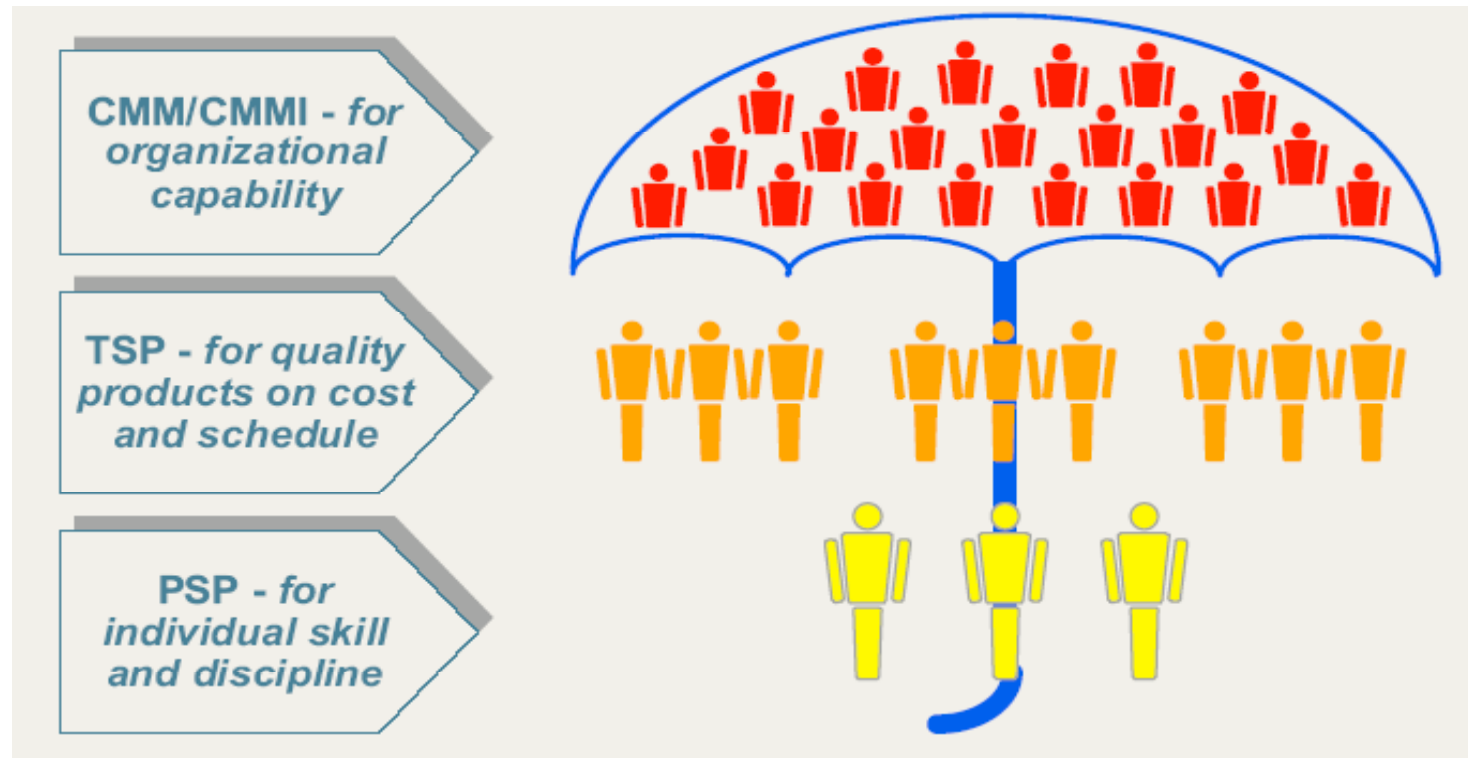
Etapas en la Mejora Continua de Procesos





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, enmarcados en el contexto de una organización que aplica CMMI,



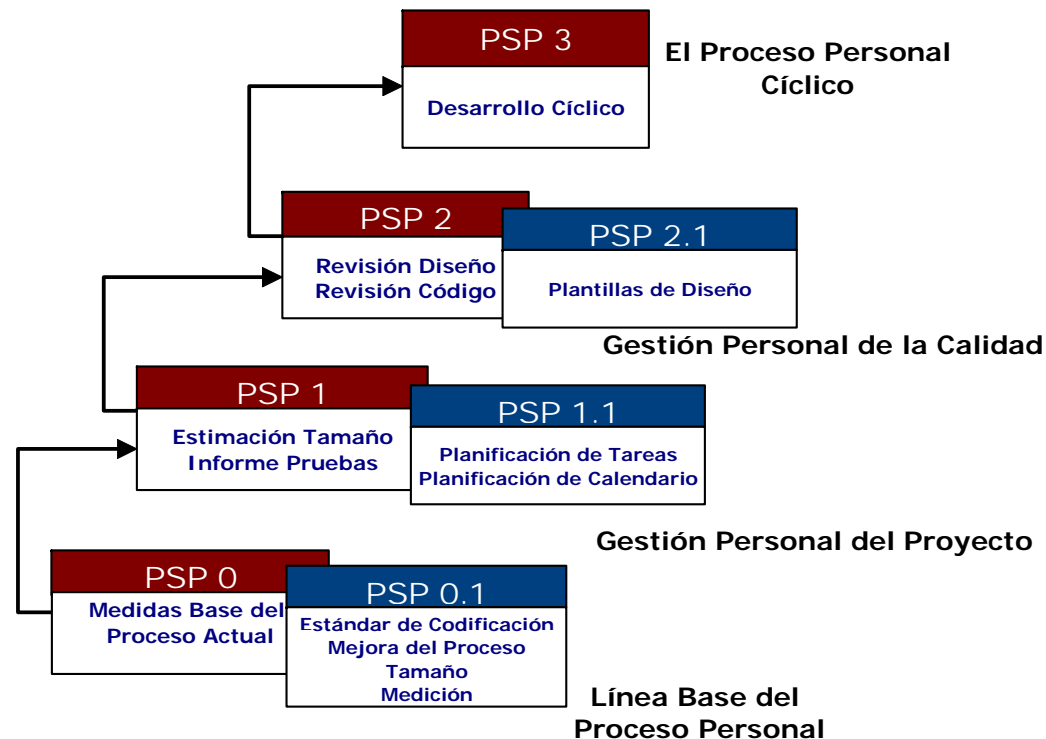


Evaluación y Mejora de Procesos – PSP Y TSP

- **PSP (Personal Software Process)**

Humfrey, W.S. (2001): Introducción al Proceso Software Personal (PSP). Addison-Wesley.

- Proporciona una serie de principios al ingeniero para llevar a cabo un proceso personal disciplinado.
- 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 INTERRUP.	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	

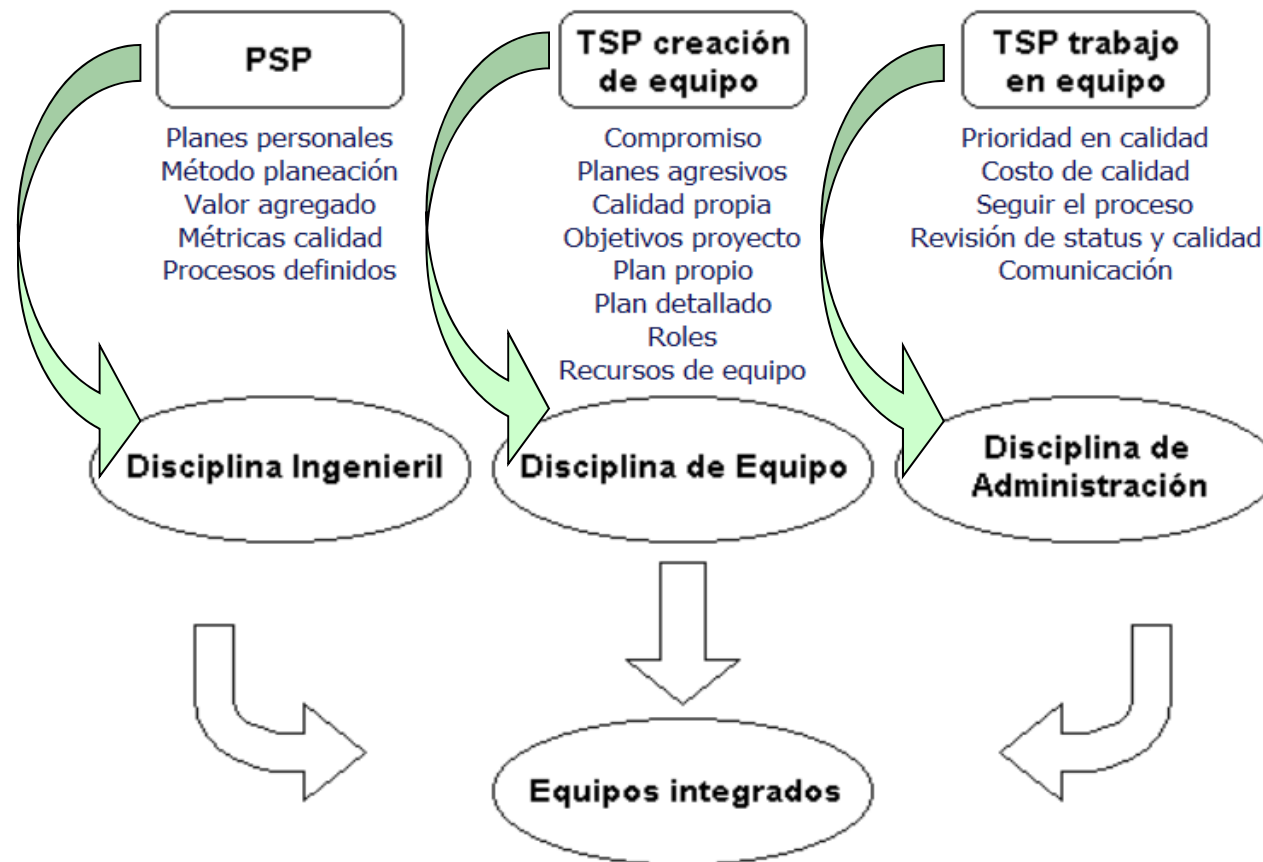
Ejemplo de formulario: registro de tiempos



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.





Medición del Software - Introducción

*"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

*"No se puede predecir lo que no se puede **medir**."*

Norman Fenton

*"Lo que no sea **medible**, hazlo medible."*

Galileo Galilei

*"No se puede controlar lo que no se puede **medir**."*

Tom De Marco

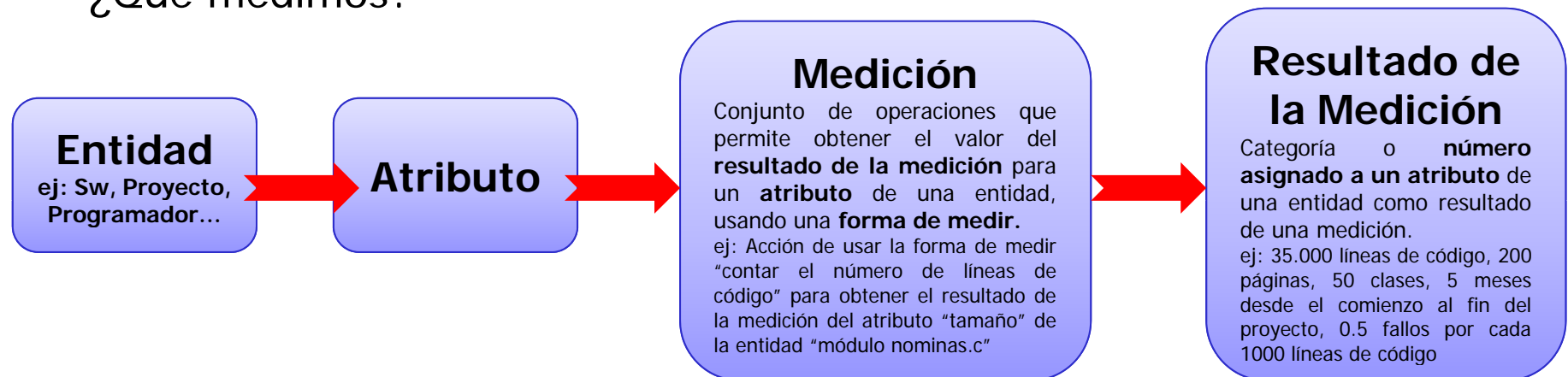
*"No todo lo que **importa** puede **medirse** fácilmente. No todo lo que puede medirse importa realmente."*

Albert Einstein



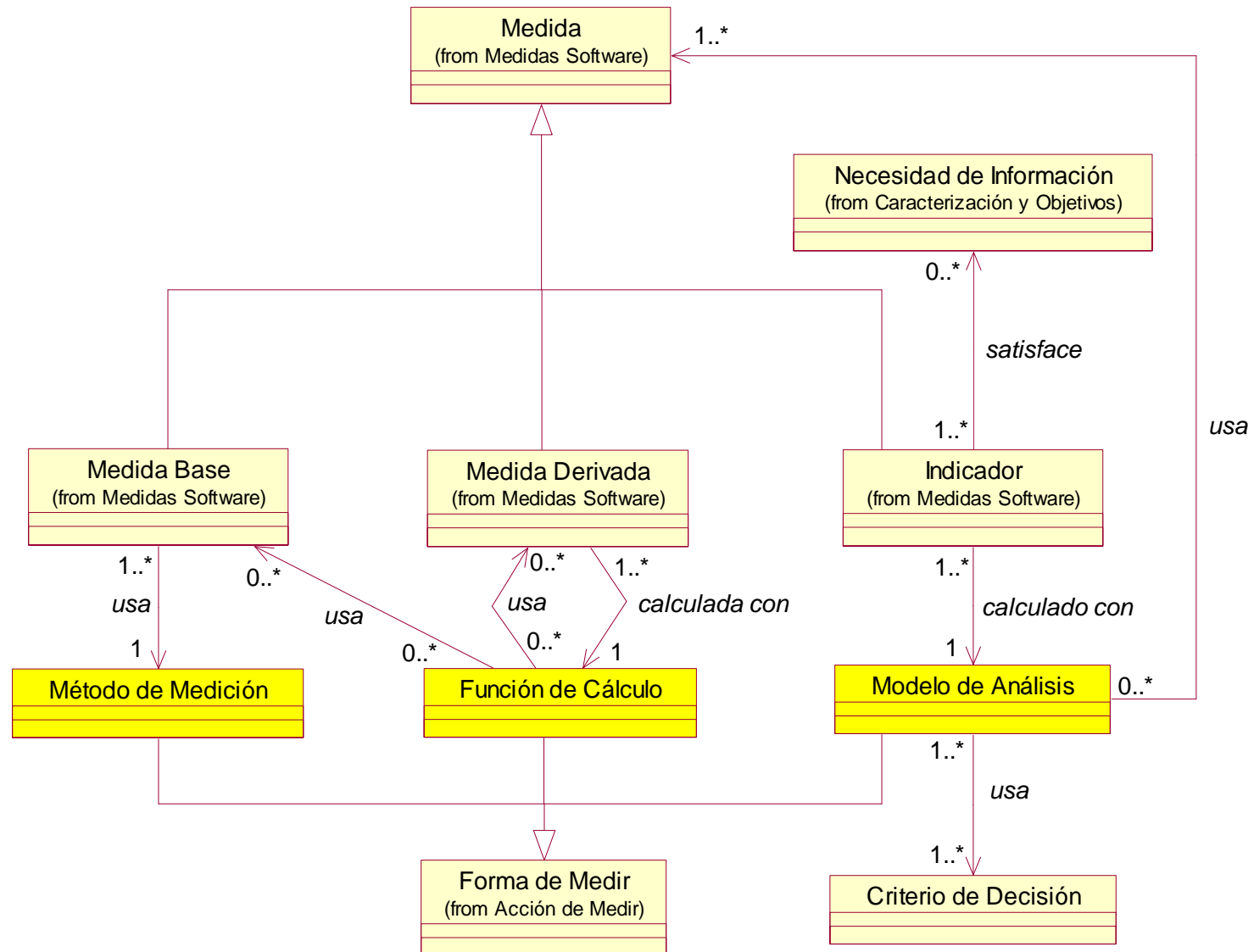
Medición del Software - Introducción

- 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):
 1. Entender qué ocurre durante el desarrollo y el mantenimiento.
 2. Controlar qué es lo que ocurre en los proyectos.
 3. Mejorar los procesos y los productos.
- ¿Qué medimos?





Medición del Software – Tipos y Formas de Medir





Medición del Software – Definiciones [SW MEASUREMENT ONTOLOGY]

Concepto	Definición	Ejemplo
Atributo	Una propiedad mensurable, física o abstracta, que comparten todas las entidades de una cierta categoría de entidad.	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	LCF (líneas de código fuente escritas). HPD (horas-programador diarias). CHP (coste por hora-programador, en unidades monetarias).
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)

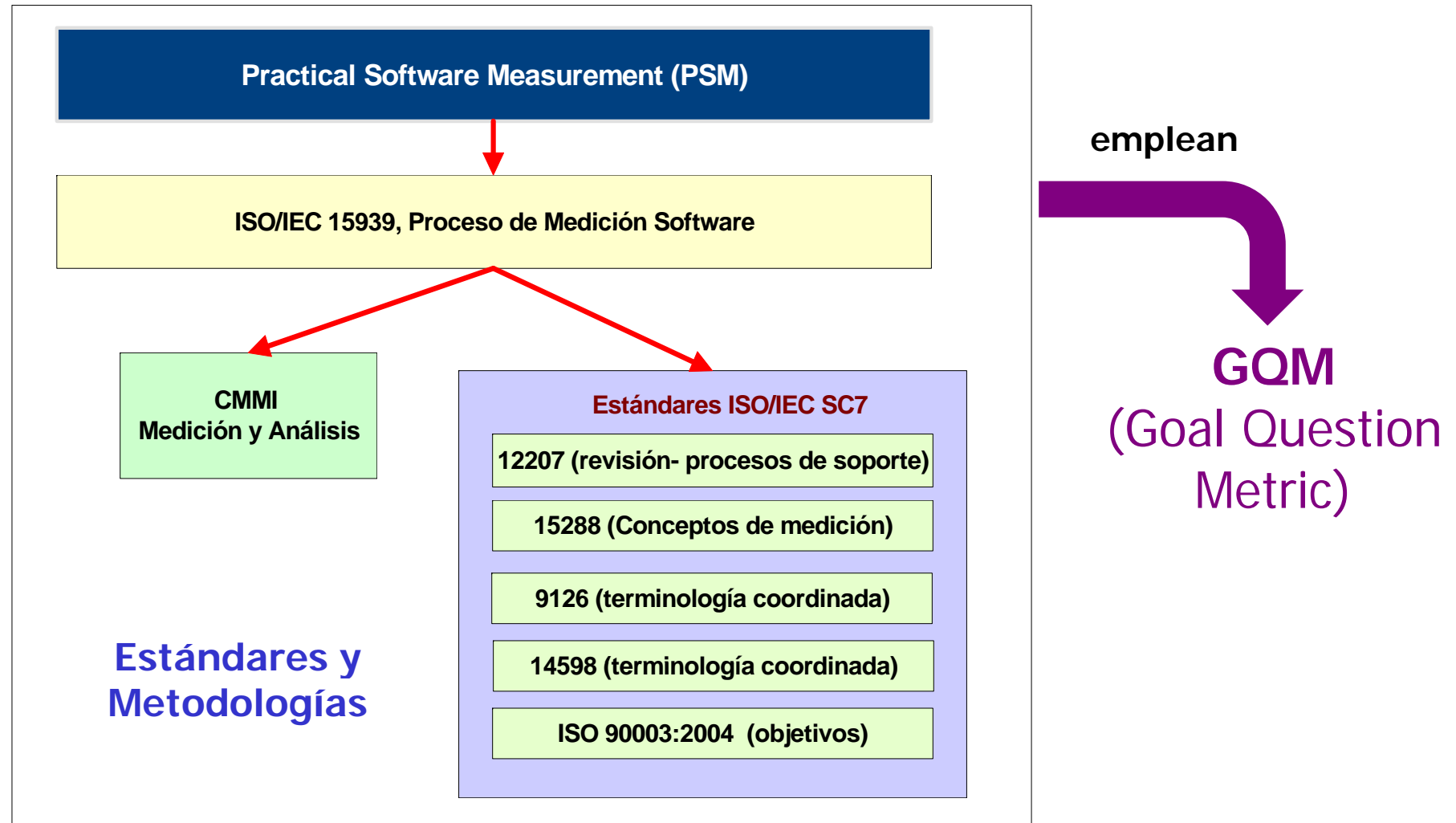


Medición del Software – Definiciones [SW MEASUREMENT ONTOLOGY]

Concepto	Definición	Ejemplo
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 .
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 .	

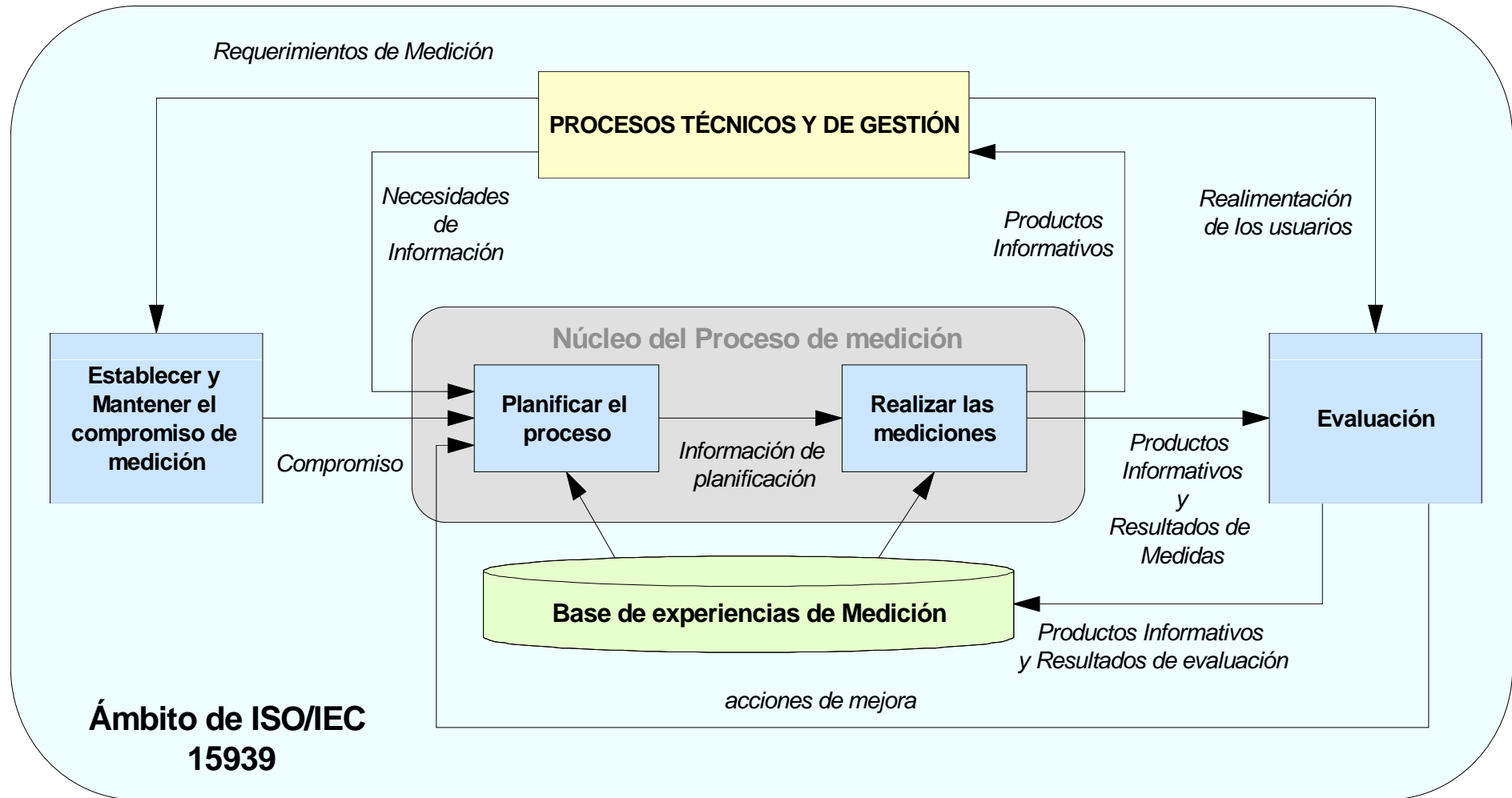


Medición del Software – Proceso de Medición ISO/IEC 15939





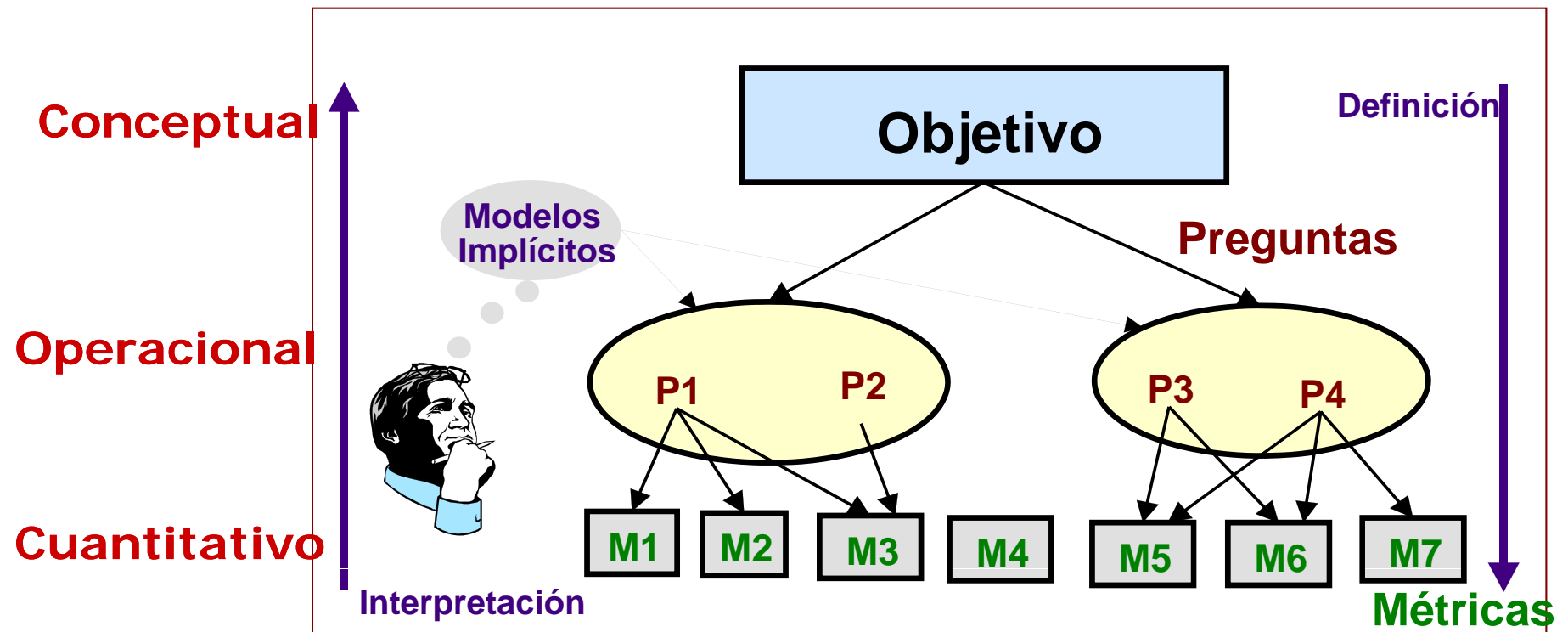
Medición del Software – Proceso de Medición ISO/IEC 15939





Medición del Software - GQM

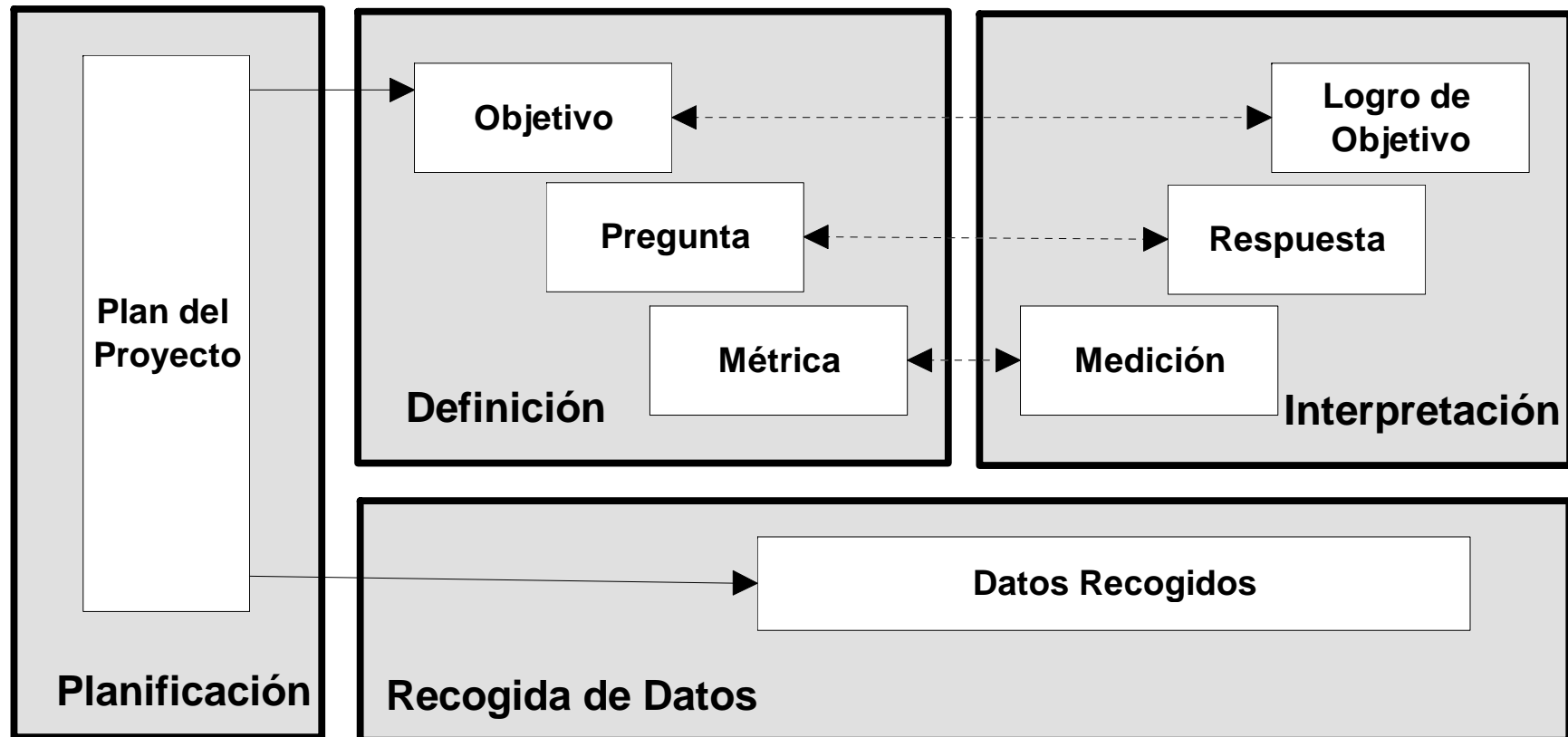
- **Principio básico:** la medición debe ser realizada, siempre, orientada a un **objetivo**.
- **GQM** define un objetivo, refina este objetivo en preguntas y define medidas/métricas que intentan dar información para responder a estas preguntas.





Medición del Software - GQM

- Fases GQM:**





Medición del Software - GQM

- **Ejemplo** - Medida para BBDD Relacionales:
 - **Objetivo GQM**
 - *Analizar* BBDD Relacionales
 - *Con el propósito de* Asegurar
 - *Con respecto a* la Mantenibilidad
 - *Desde el punto de vista de* los Diseñadores de BBDD
 - *En el contexto de* 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

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

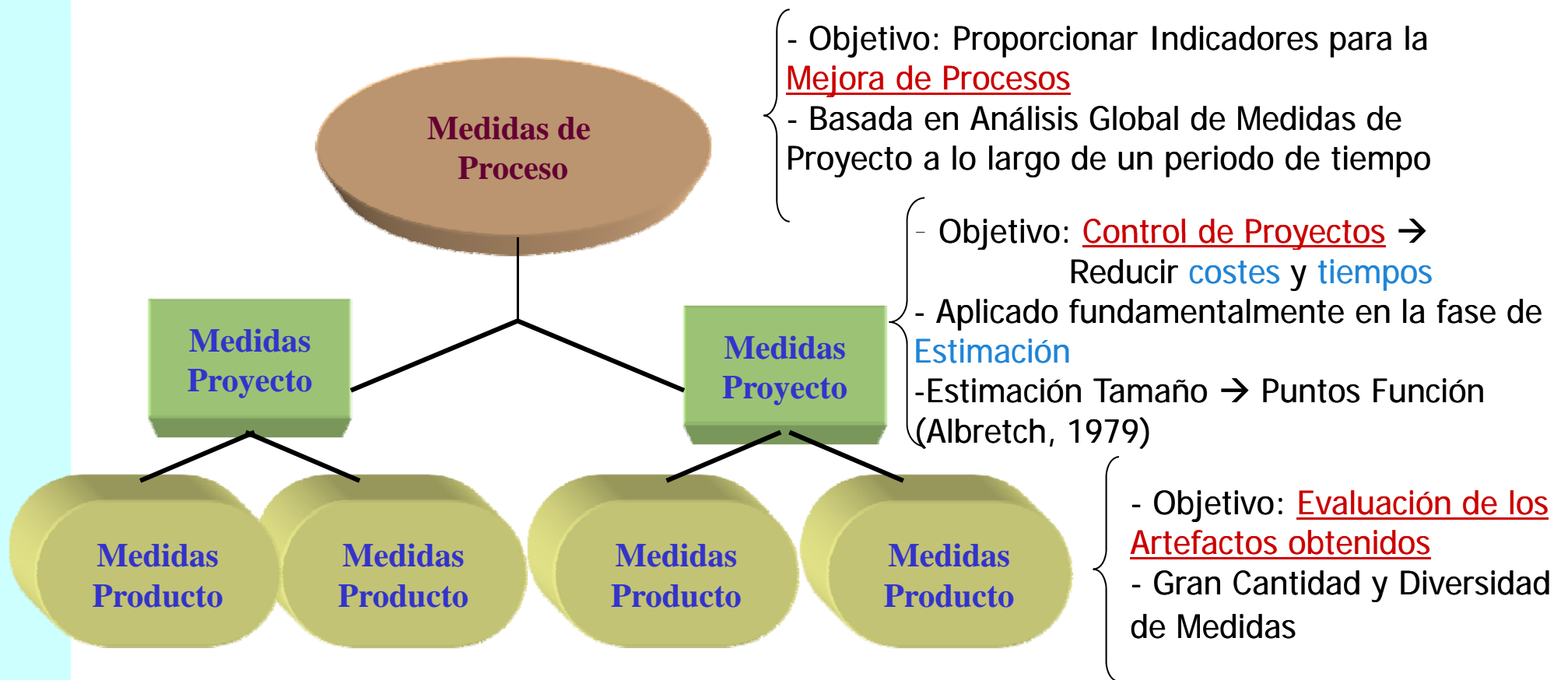
- **Pregunta 2**

- 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

- **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.

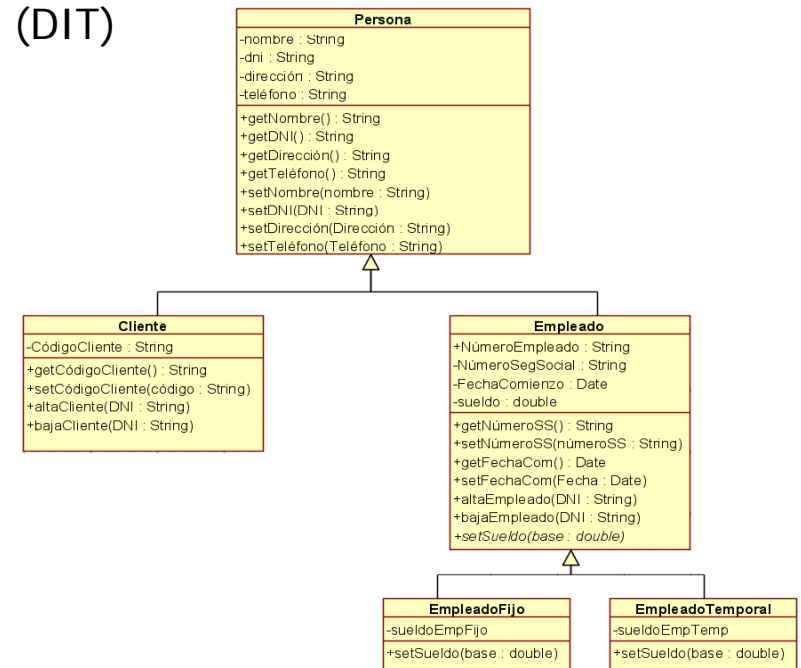
- **Ejemplos de Medidas para Sistemas OO:**

- Métodos Ponderados por Clase (WMC) $WMC = \sum_{i=1}^n n_i C_i$
- Profundidad del Árbol de Herencia de una Clase (DIT)
- Número de Hijos (NOC)

WMC(Persona) = 8

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

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

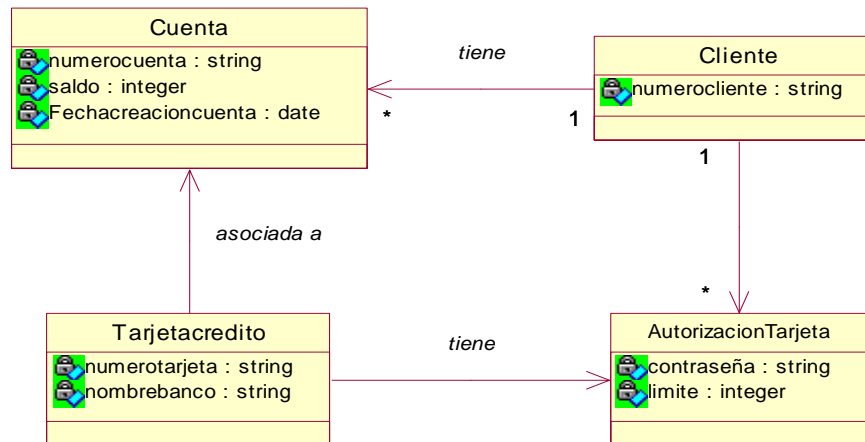




Medición del Software - Medidas

- Ejemplos de Medidas para Sistemas OO:

- Acoplamiento entre Objetos (CBO)



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

- Respuesta para una Clase (RFC)

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

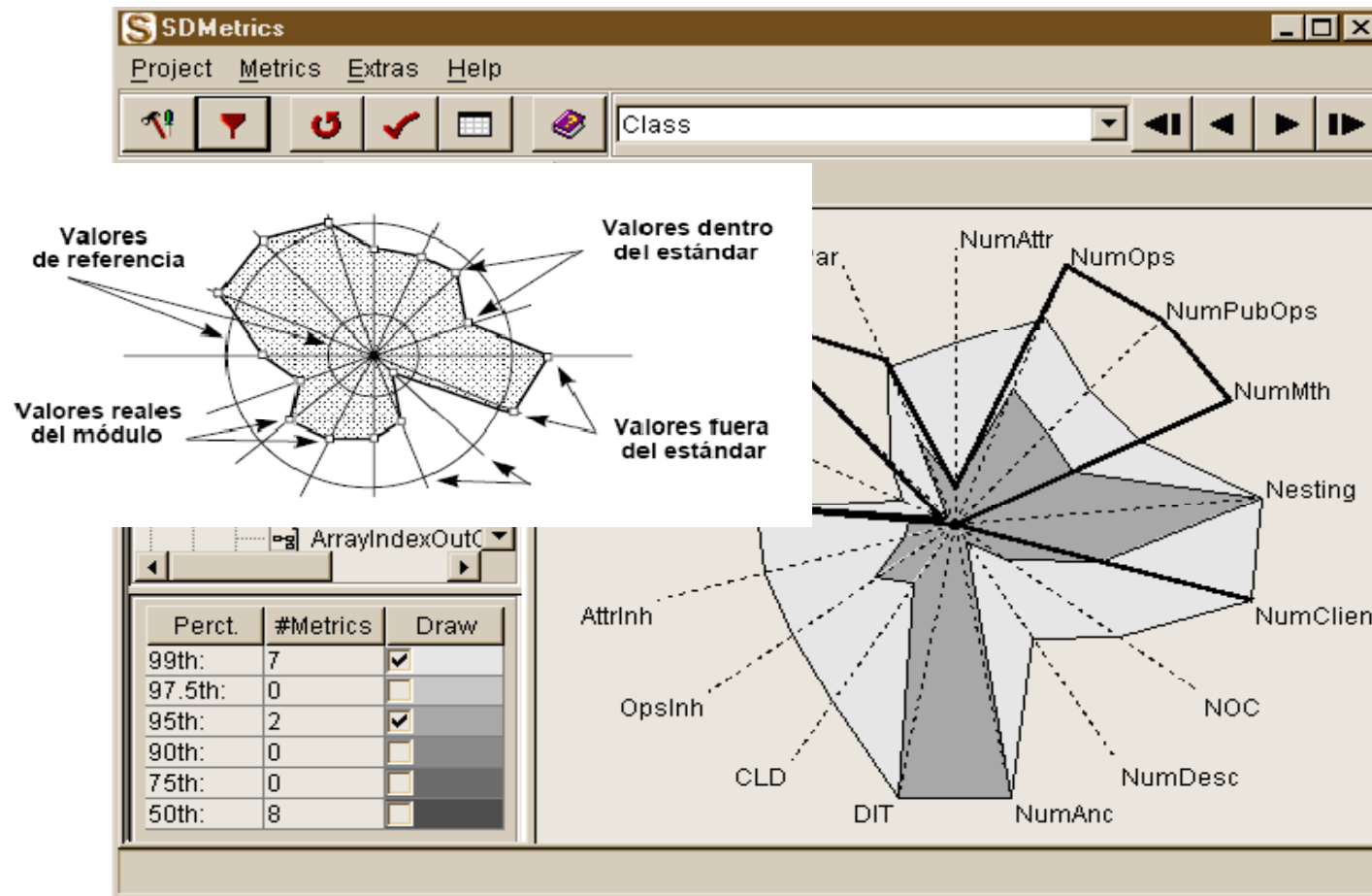
$$RS = \{ A::f1, A::f2, A::f3, A::f4 \} \cup \{ B::f1, B::f2, C::f3 \} \cup \{ B::f1 \} \cup \{ A::f4, B::f3, C::f1, C::f2 \} = \{ A::f1, A::f2, A::f3, A::f4, B::f1, B::f2, B::f3, C::f1, C::f2, C::f3 \}$$

RFC(A)=10



Medición del Software – Diagramas de Kiviat

A través de mapas de valores de distintas métricas del producto y sus tolerancias obtenemos una “foto” del índice de calidad del mismo.





Gestión de la Calidad – Sw Quality Management (SQM)

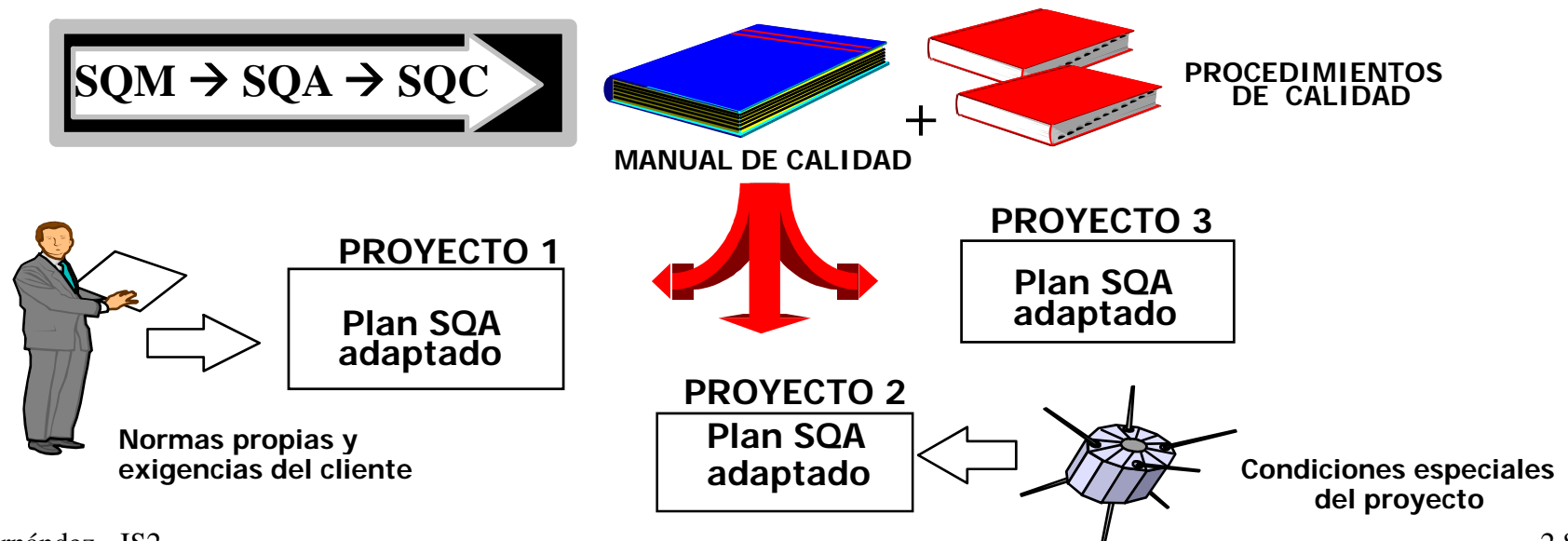
- Es **aplicable a** los productos, procesos y recursos implicados en el desarrollo de Sw, pudiendo emplearse tanto para productos intermedios como para el producto final. Los **procesos** SQM tienen como objetivos:
 - Satisfacer los requisitos de clientes y demás stakeholders, buscando su satisfacción y alcanzar la la calidad del software necesaria .
 - Instaurar una cultura de calidad en la organización basada en la responsabilidad de los distintos participantes en el proceso de desarrollo.
- Los principales procesos SQM son:
 - **Aseguramiento de Calidad** (*Quality Assurance*)
 - **Verificación** (*Verification*)
 - **Validación** (*Validation*)

} (*Quality Control*)



Gestión de la Calidad - Sw Quality Assurance (SQA)

- 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 - Sw Quality Control (SQC)

- **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**.
- **Actividades Verificación:**
 - ¿Estamos **construyendo correctamente** el producto?
 - El software debe ser conforme a su especificación.
 - Objetivo: Demostrar la consistencia, compleción y corrección de los **artefactos software** entre las **fases** del ciclo de desarrollo de un proyecto.
 - Técnica más utilizada: **Revisiones SW**
- **Actividades 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**



Verificación y Validación – Tipos 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**.

En cualquier caso pueden ser Técnicas o de Gestión

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

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



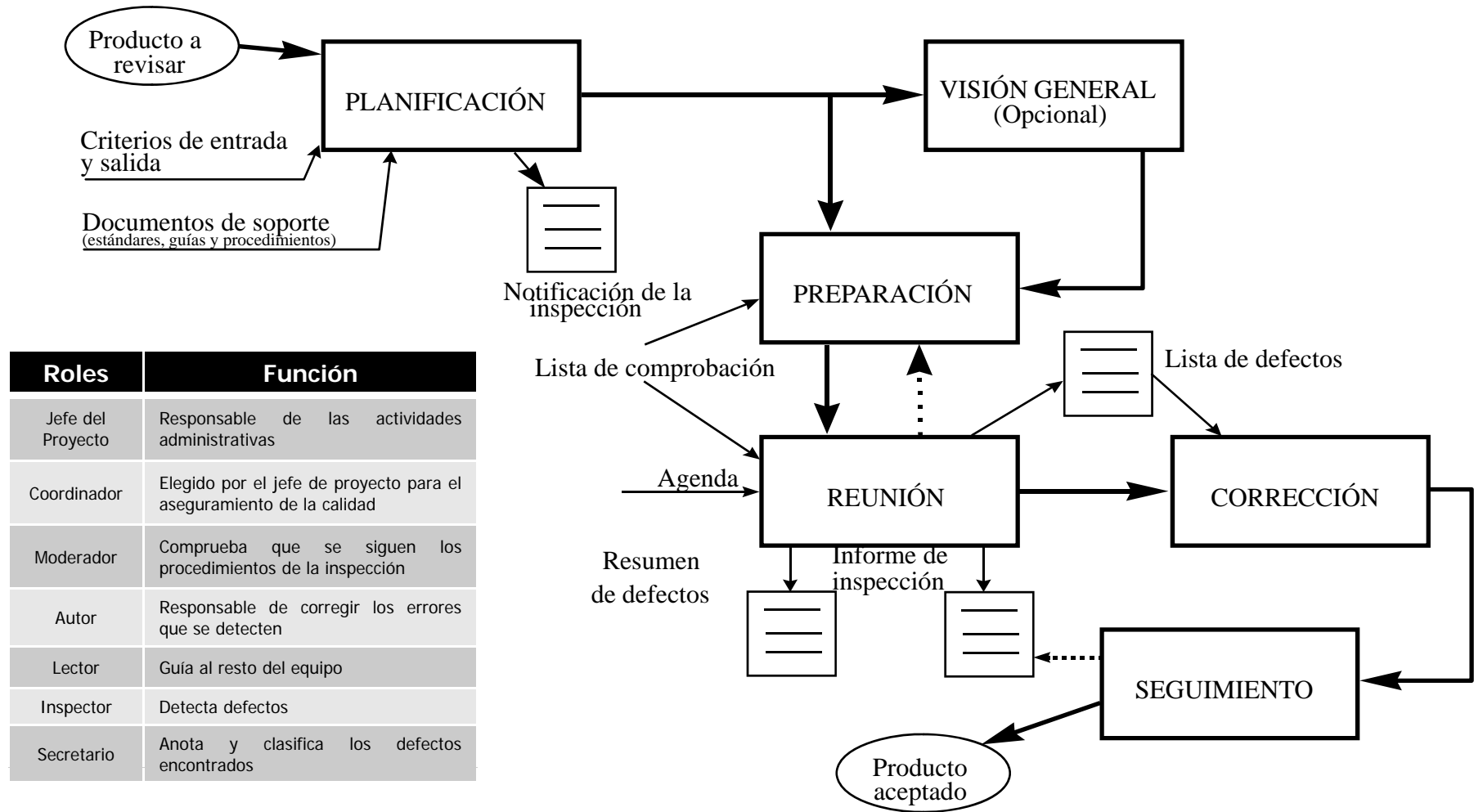
Verificación y Validación - Inspecciones

- **Inspecciones**, ¿qué son?
 - Orientadas a la detección de **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

- Inspecciones.** Flujo de Trabajo y Roles.





- Proporcionan información del progreso o resultados de la inspección

- Notificación de la reunión de inspección → Anuncio formal

- Localización
- Descripción → Problema
- Clasificación

- Informe de Inspección

Juan Hernández - IS2



Verificación y Validación - Inspecciones

- **Inspecciones.** Checklist de Comprobación.

LISTA DE COMPROBACIÓN PARA LA REVISIÓN DE UNA ERS

Organización y compleción

- ¿Son correctas las referencias cruzadas a otros requisitos?
- ¿Están todos los requisitos escritos de una manera consistente y a un nivel apropiado de detalle?

...

Corrección

- ¿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?

...

Atributos de Calidad

- ¿Están especificados correctamente todos los requisitos de rendimiento?
- ¿Están correctamente especificadas todas las consideraciones de seguridad?

Trazabilidad

- ¿Está cada requisito identificado de una manera única y correcta?
- ¿Es trazable cada requisito funcional a otros requisitos de más alto nivel?

Otros aspectos

- ¿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

Descomposición en paquetes

- ¿Está definido un nombre único por cada paquete y existe una descripción breve?
- El nombre del paquete ¿es claro y fácilmente entendible?

...

Actores

- ¿Están definidos todos los actores del sistema?
- ¿Es correcta la definición de cada actor (cada actor define un único rol)?

...

Casos de Uso

- ¿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?

Diagramas de Casos de Uso

- 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 (*uses*, *extends*)?
- ¿Son correctas las relaciones de los actores con los casos de uso?



Verificación y Validación - Walkthroughs

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

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



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.



Verificación y Validación - Auditorías

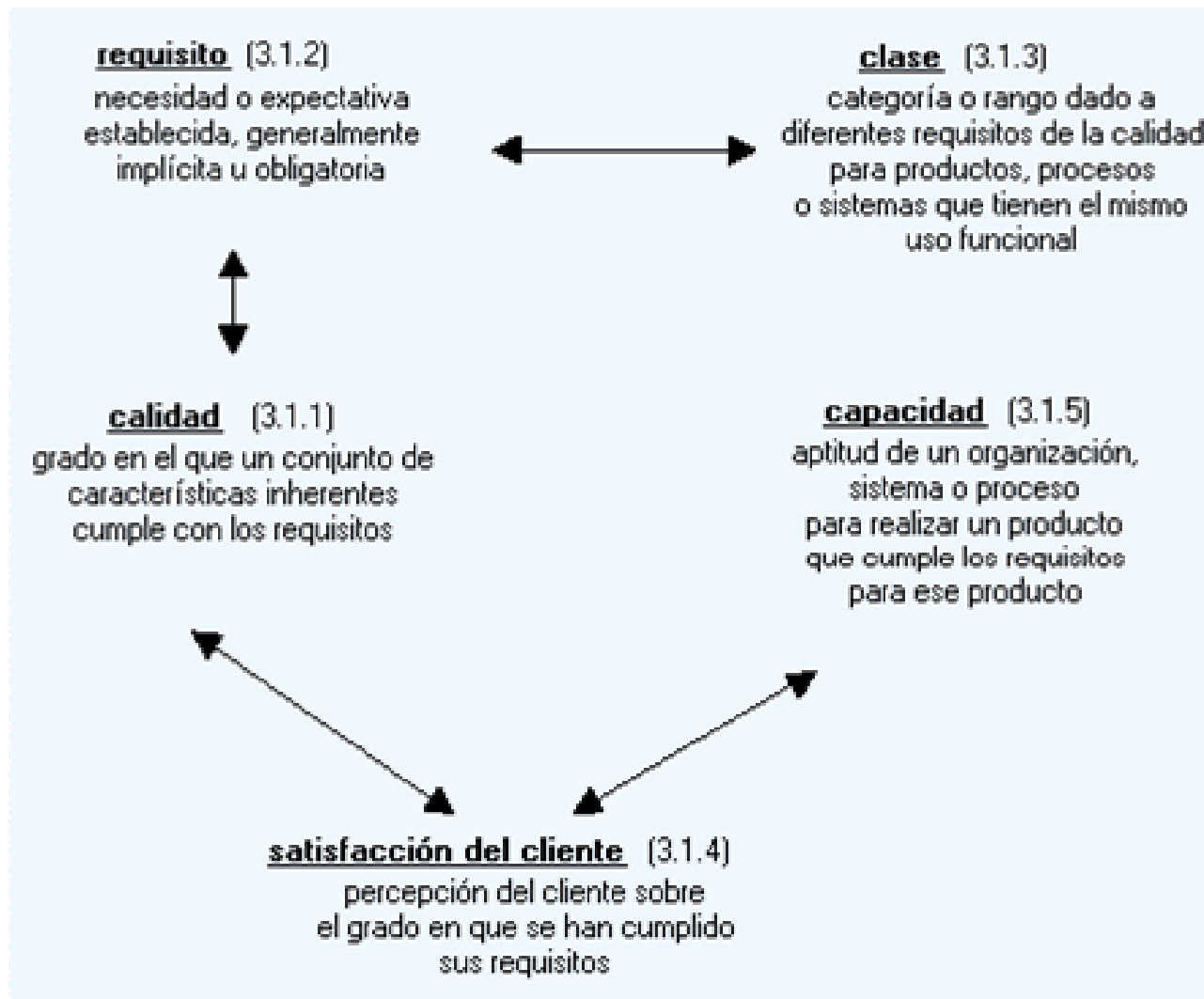
- Auditorías vs Revisiones:

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.



APÉNDICE A – Conceptos ISO 9000:2005

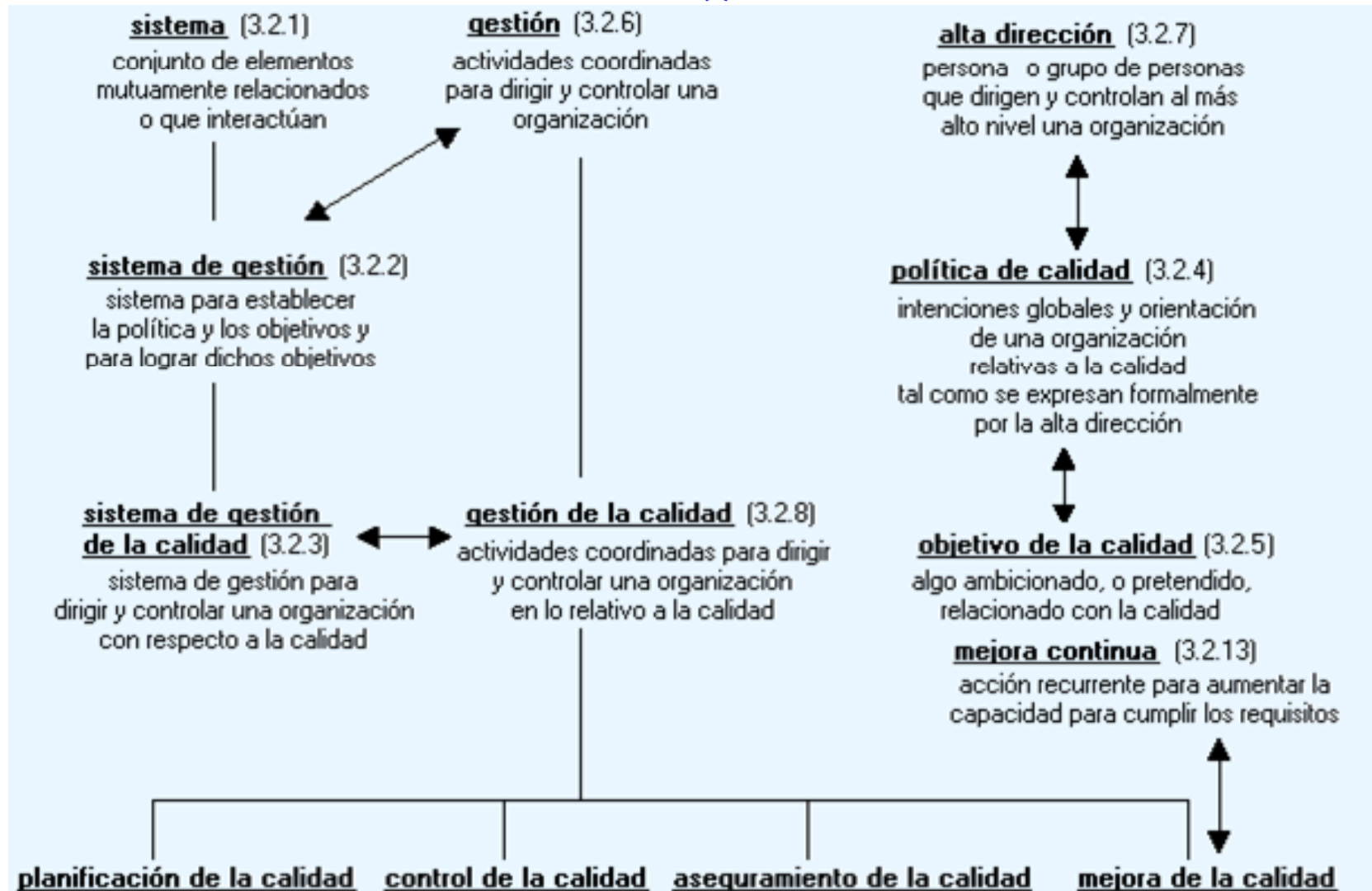
Términos relativos a la Calidad





APÉNDICE A – Conceptos ISO 9000:2005

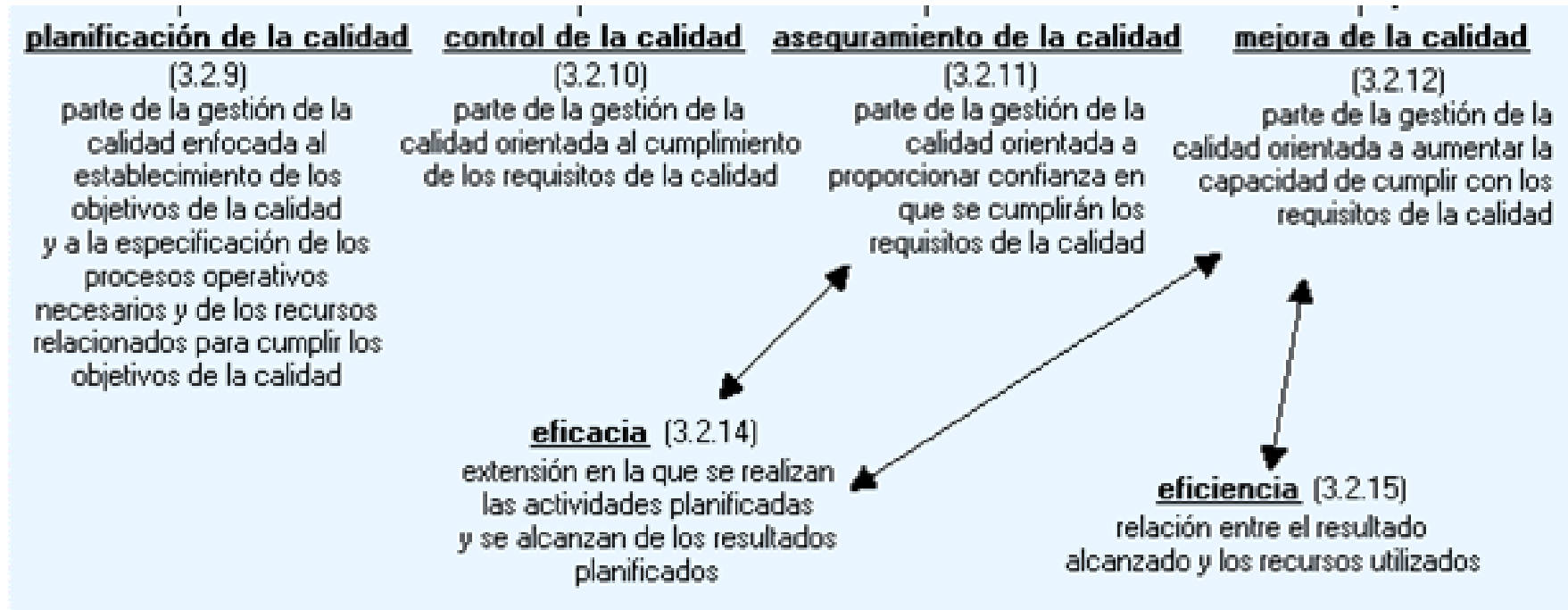
Términos relativos a la Gestión (i)





APÉNDICE A – Conceptos ISO 9000:2005

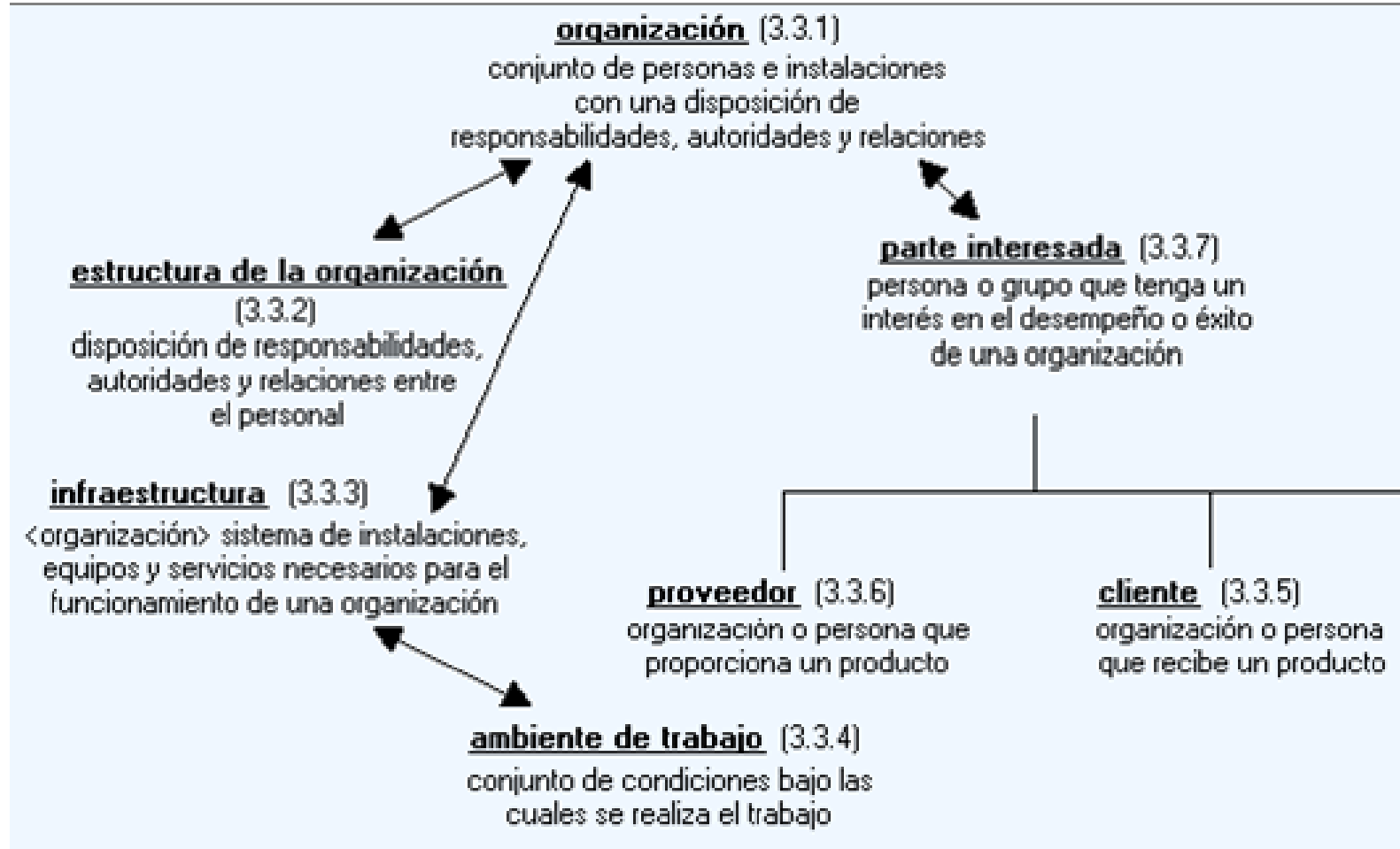
Términos relativos a la Gestión (ii)





APÉNDICE A – Conceptos ISO 9000:2005

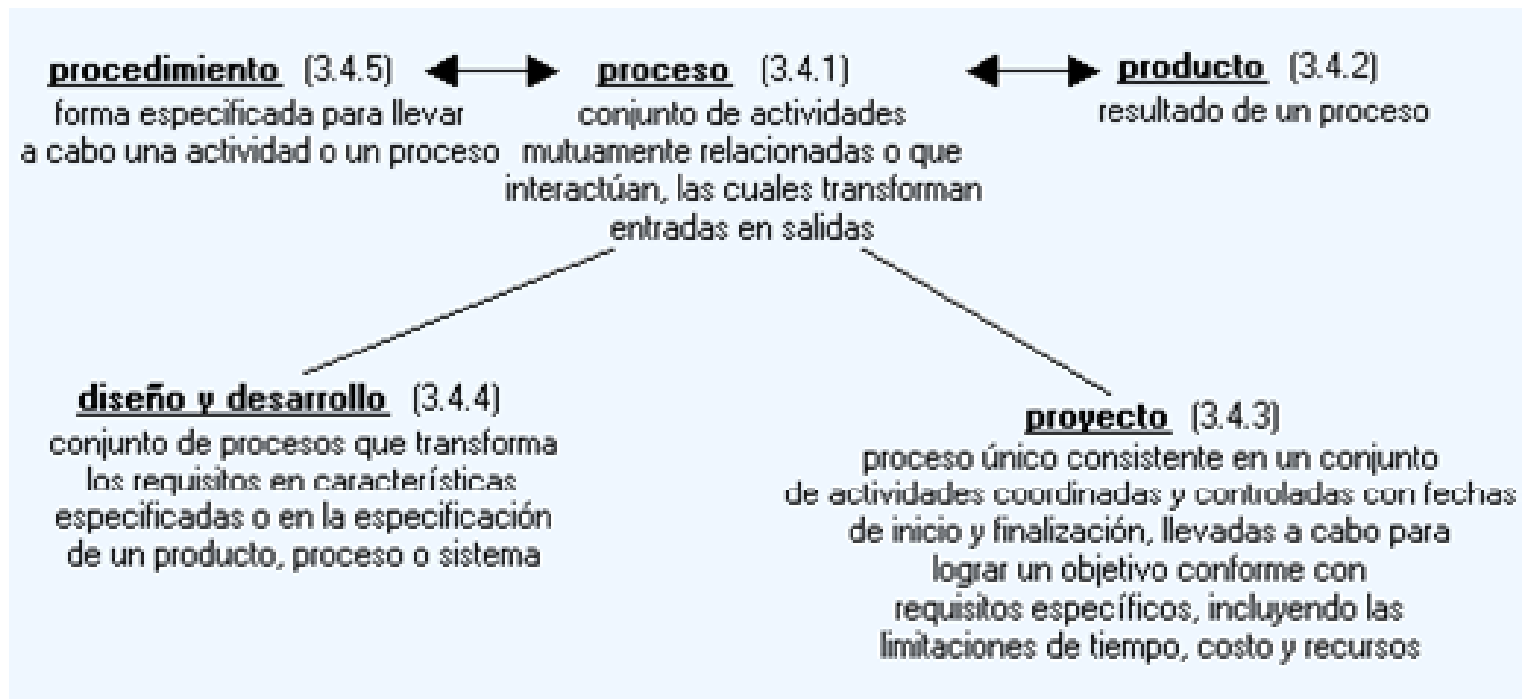
Términos relativos a la Organización





APÉNDICE A – Conceptos ISO 9000:2005

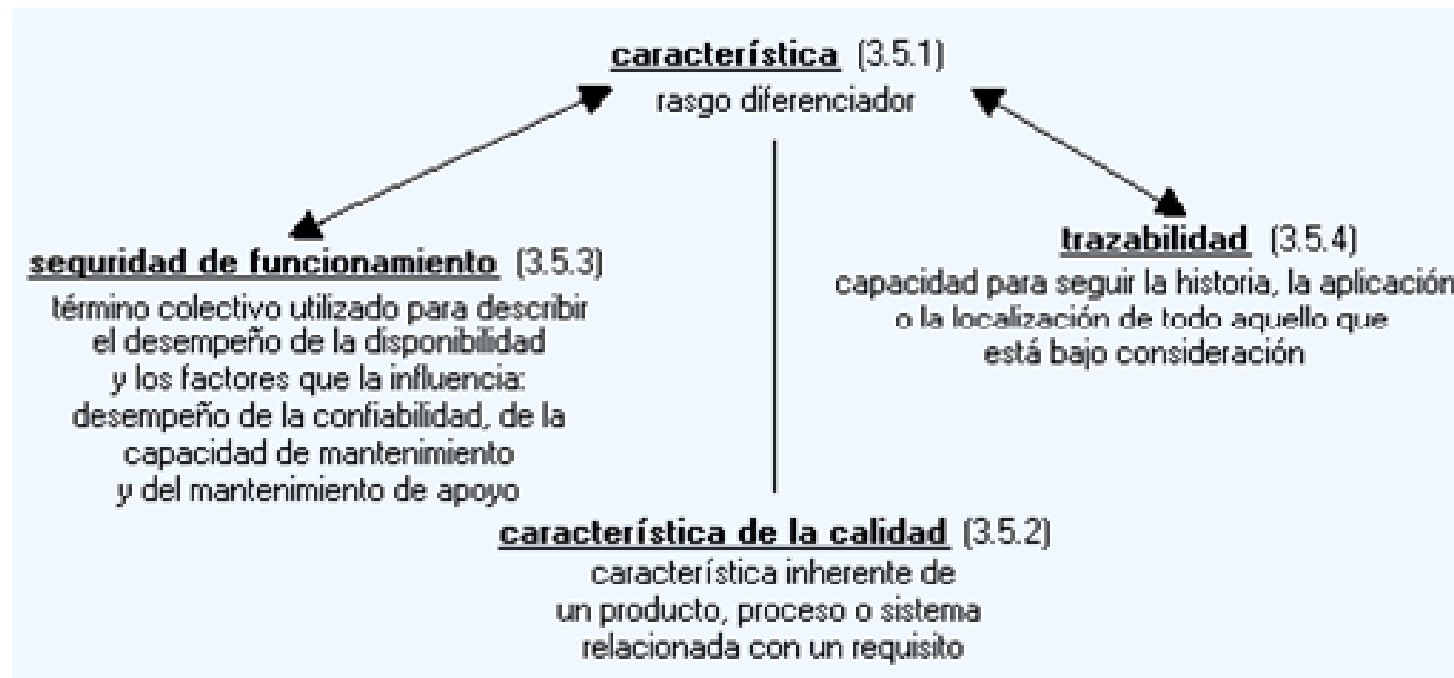
Términos relativos al Proceso y al Producto





APÉNDICE A – Conceptos ISO 9000:2005

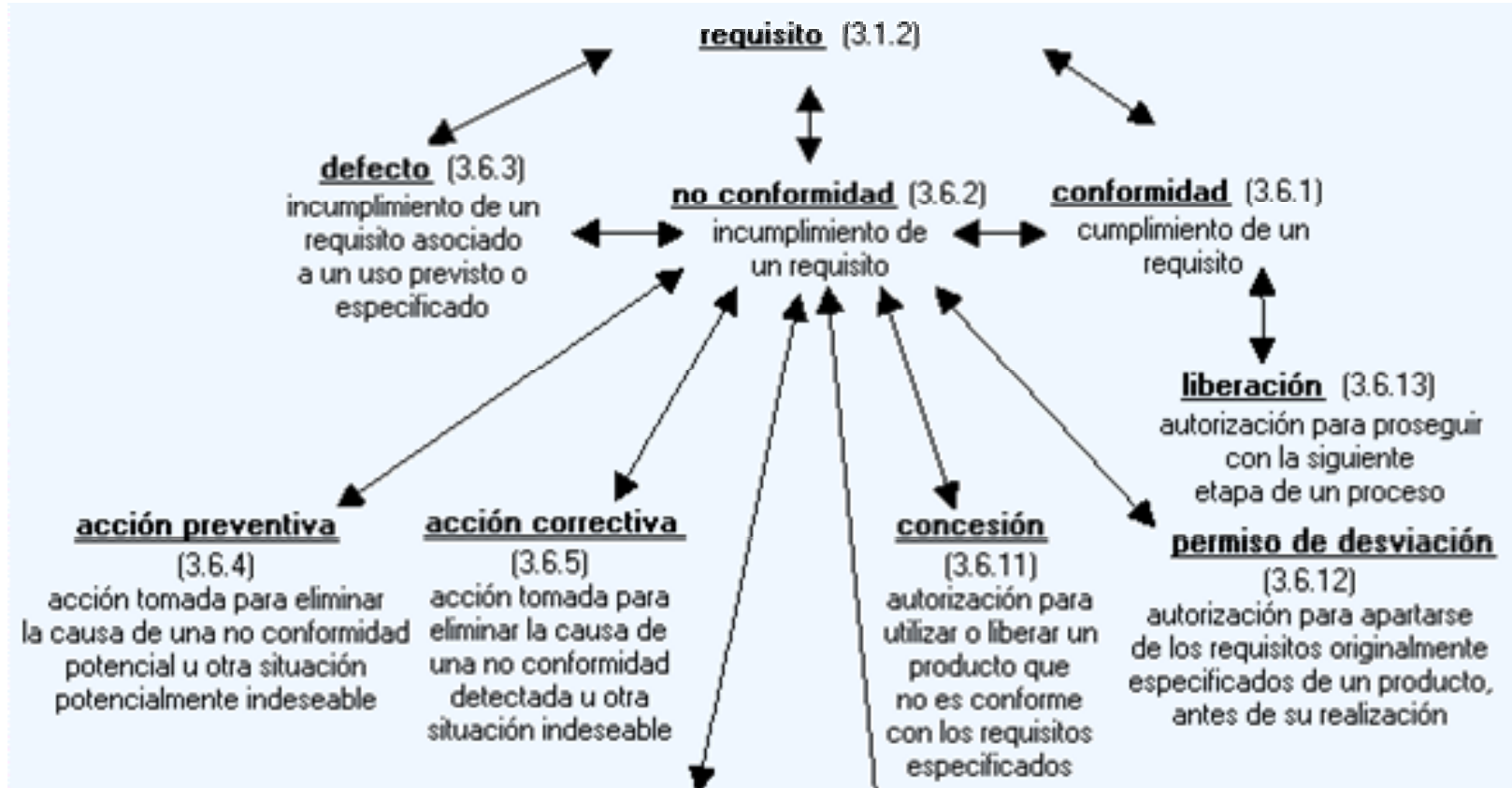
Términos relativos a las Características





APÉNDICE A – Conceptos ISO 9000:2005

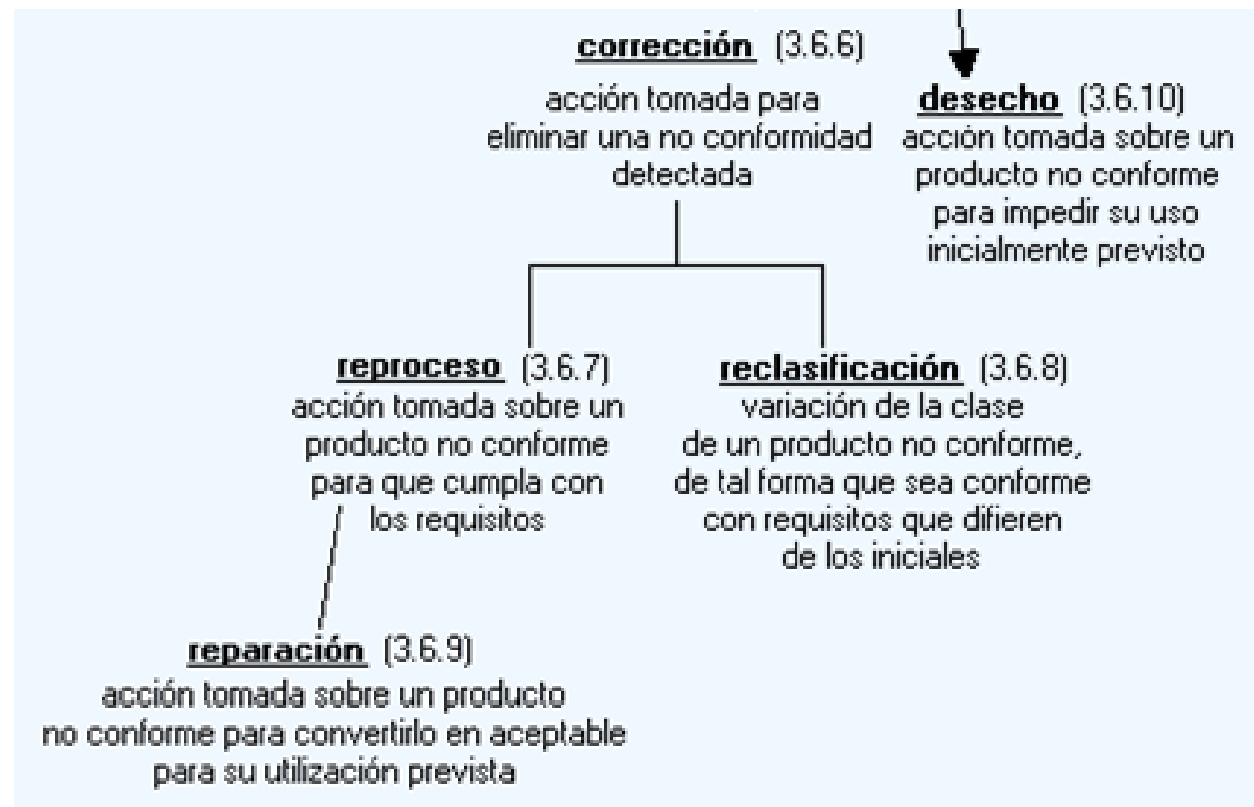
Términos relativos a la Conformidad (i)





APÉNDICE A – Conceptos ISO 9000:2005

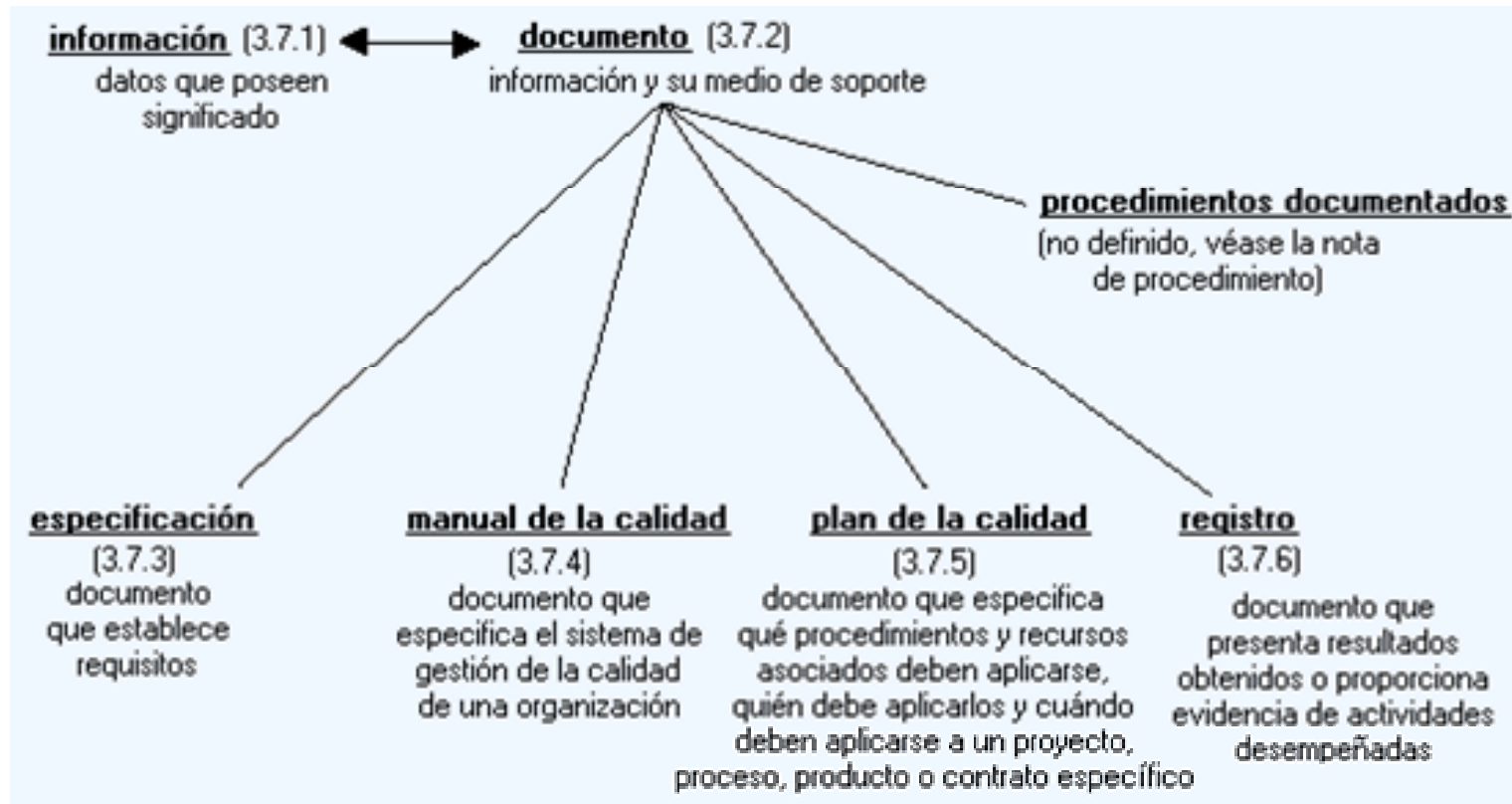
Términos relativos a la Conformidad (ii)





APÉNDICE A – Conceptos ISO 9000:2005

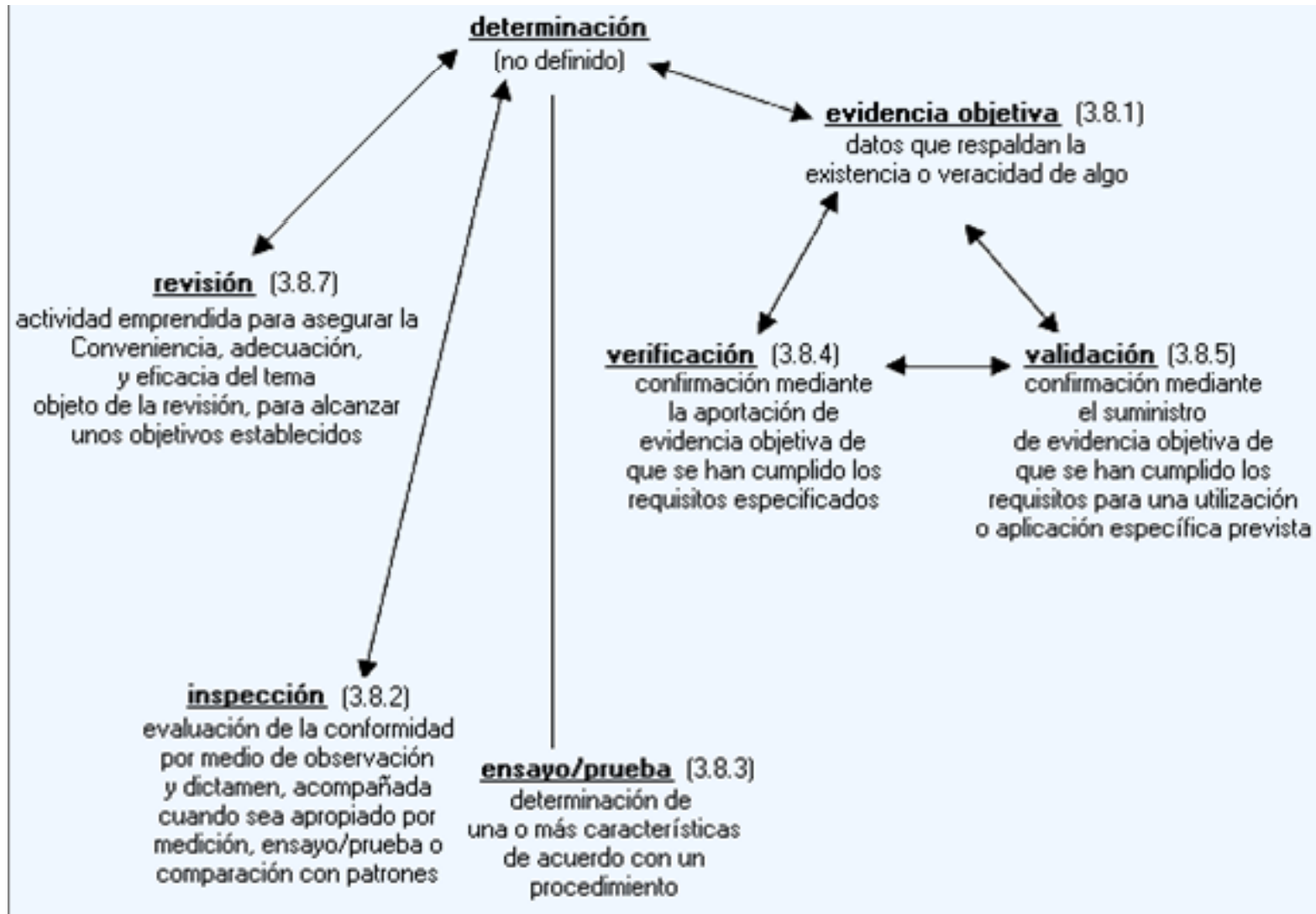
Términos relativos a la Documentación





APÉNDICE A – Conceptos ISO 9000:2005

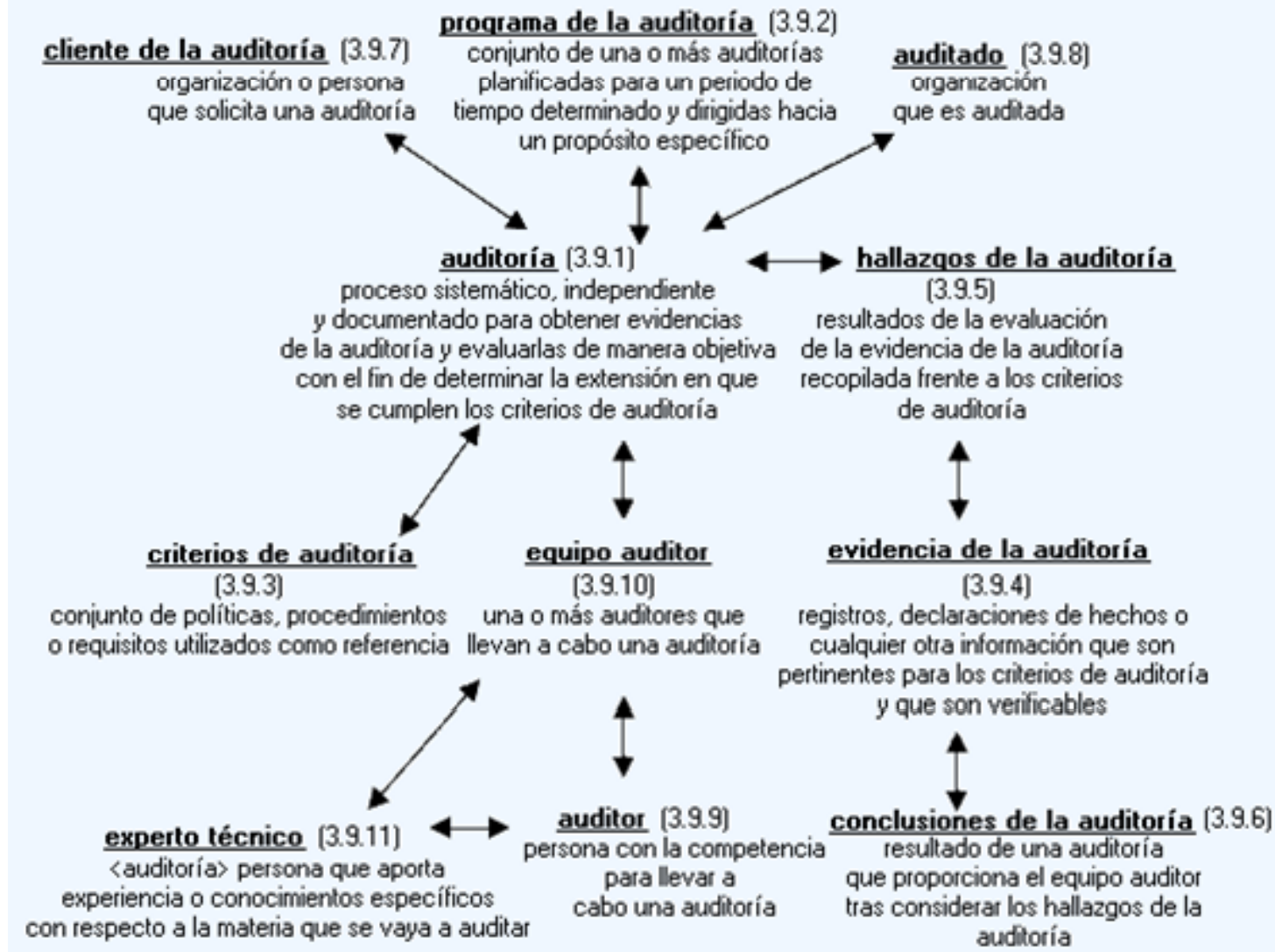
Términos relativos al Examen





APÉNDICE A – Conceptos ISO 9000:2005

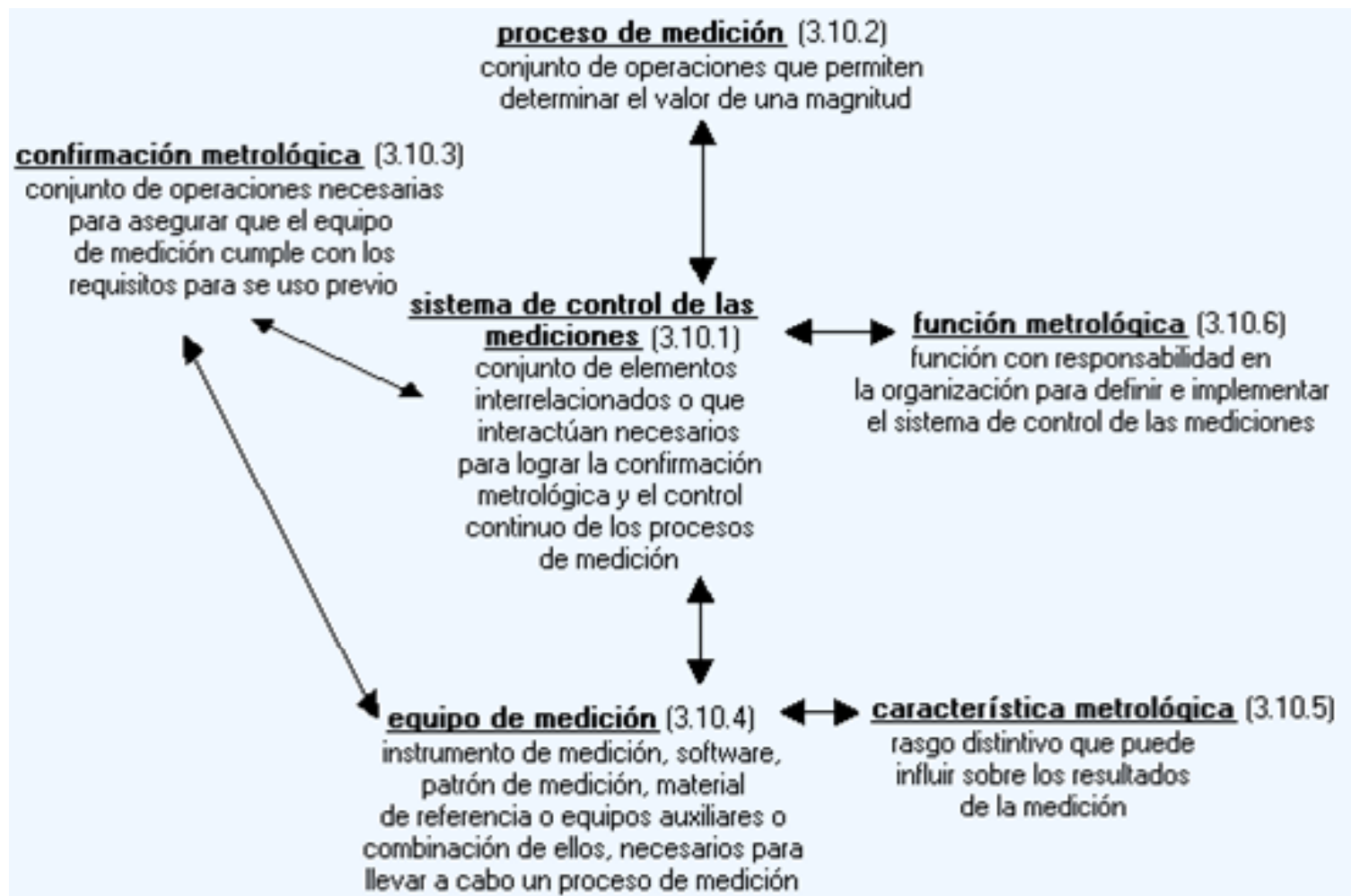
Términos relativos a la Auditoría





APÉNDICE A – Conceptos ISO 9000:2005

Términos relativos al Aseguramiento de la Calidad para los Procesos de Medición





APÉNDICE B 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.



APÉNDICE B 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.



APÉNDICE B 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.



APÉNDICE C Lista de Defectos de Inspecciones

- Clasificación: Factor de Calidad, Tipo, Clase, Gravedad.

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



APÉNDICE C Lista de Defectos de Inspecciones

- Clasificación: Factor de Calidad, Tipo, Clase, Gravedad.

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