



INGENIERÍA DE SOFTWARE

GCS
MÉTRICAS
CALIDAD DE SOFTWARE

EN CLASES ANTERIORES VIMOS ...

Conceptos generales

Modelos proceso

Metodologías ágiles

Desarrollo de Software Dirigido por Modelos

Problemas de Comunicación

Elicitación de requerimientos

Técnicas de elicitación de requerimientos

Definición de Requerimientos

- Funcionales
- No Funcionales

Ingeniería de Requerimientos

Técnicas de especificación de requerimientos

EN CLASES ANTERIORES VIMOS ...

Definición de proyecto

- Características

Gestión de Proyecto

- Métricas / Estimaciones / Calendario temporal / Organización del personal / Análisis de riesgos / Seguimiento y control

Planificación

- Temporal
 - PERT / CPM / Gantt / PERT+CPM
- Organizativa
 - Modelo MOI / DD- DC -CC

Riesgos

- Definición
- Estrategias de riesgos
- Clasificación de riesgos
- Proceso de Gestión de Riesgos
 - 1 Identificación de Riesgos
 - 2 Análisis de Riesgos
 - 3 Planeación
 - 4 Supervisión

GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»¿Qué es Software?

- Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación

»Gestión de Configuración es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

»Es una actividad de autoprotección que se aplica durante el proceso del software.



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»El resultado del proceso de Software se puede dividir en:

- Programas (códigos y ejecutables)
- Documentos
- Datos

Elementos de la
configuración (ECS)



ECS - Cambian
constantemente



Control muy exhaustivo
de esos cambios



GCS



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

Plan del proyecto software

- a) Especificación de requerimientos del software
- b) Prototipo ejecutable o en papel

Manual de usuario preliminar

Especificación de diseño:

- a) Diseño preliminar
- b) Diseño detallado

Listados del código fuente

- a) Planificación y procedimiento de prueba
- b) Casos de prueba y resultados registrados

Manuales de operación y de instalación

Programas ejecutables

Manual de usuario

Documentos de mantenimiento

- a) Informes de problemas del software
- b) Peticiones de mantenimiento
- c) Órdenes de cambios de ingeniería

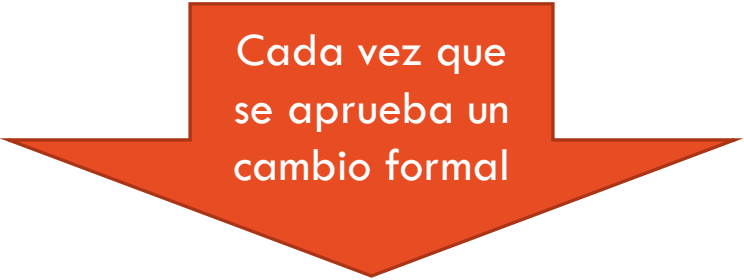
Estándares y procedimientos de ingeniería del software



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»El cambio se puede producir en cualquier momento, las actividades de la GCS sirven para:

- Identificar el cambio
- Controlar el cambio
- Garantizar que el cambio se implemente adecuadamente
- Informar del cambio a todos aquellos que puedan estar afectados



Cada vez que
se aprueba un
cambio formal

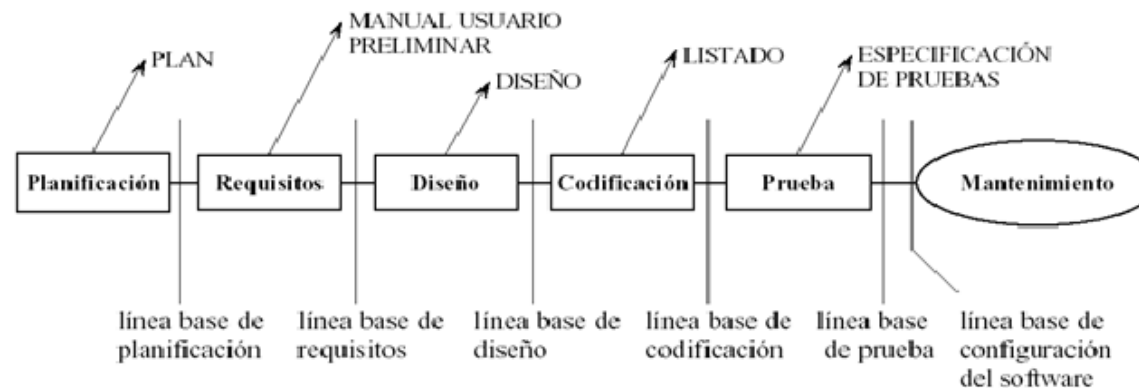
Línea Base



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Línea Base

- Una línea base es un concepto de GCS que nos ayuda a controlar los cambios
- Definición de la IEEE
 - Una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambio
- En el contexto de la Ingeniería de Software:
 - una línea base es un punto de referencia en el desarrollo del software que queda marcado por el envío de uno o mas ECS y su aprobación



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Importancia de la GCS

- ¿Cómo identifica y gestiona una organización las diferentes versiones existentes de un programa (y su documentación) de forma que se puedan introducir cambios eficientemente?
- ¿Cómo controla la organización los cambios antes y después de que el software sea distribuido al cliente?
- ¿Quién tiene la responsabilidad de aprobar y de asignar prioridades a los cambios?
- ¿Cómo podemos garantizar que los cambios se han llevado a cabo adecuadamente?
- ¿Qué mecanismo se usa para avisar a otros de los cambios realizados?



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

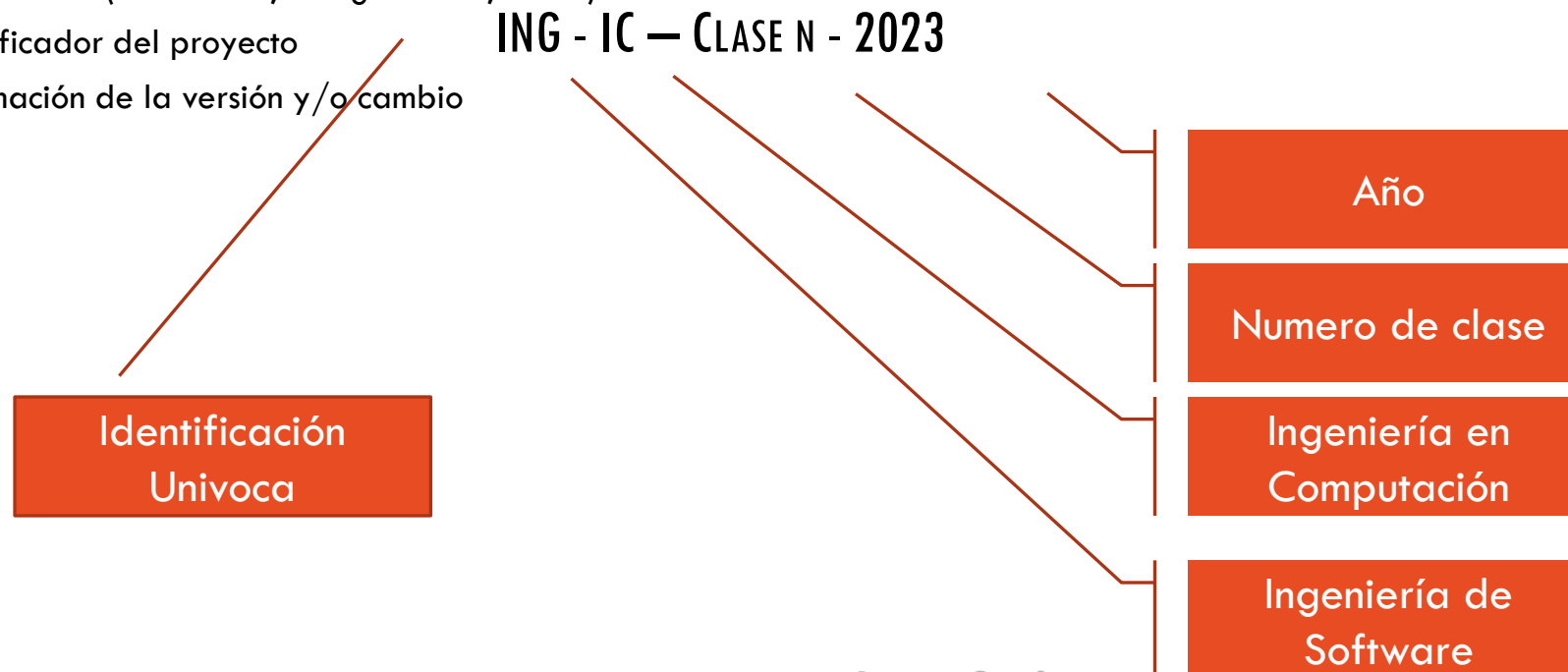
1. Identificación
2. Control de versiones
3. Control de cambios
4. Auditorías de la configuración
5. Generación de informes



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 1- Identificación de los objetos en la GCS
 - Nombre: cadena de caracteres sin ambigüedad
 - Descripción: lista de elementos de datos que identifican:
 - Tipo de ECS (documento, código fuente, datos)
 - Identificador del proyecto
 - Información de la versión y/o cambio



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 2 - Control de versiones
 - Permite al usuario especificar configuraciones alternativas del sistema mediante la selección de versiones adecuadas (por ejemplo asociando atributos que la identifican)
 - Combinación de procedimientos y herramientas para gestionar las versiones de los ECS
 - Ejemplo de versiones
 - Un programa puede contener los módulos 1-2-3-4-5
 - Una versión puede utilizar los módulos 1235
 - Otra versión puede utilizar los módulos 1245
 - Dos variantes de una misma versión



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

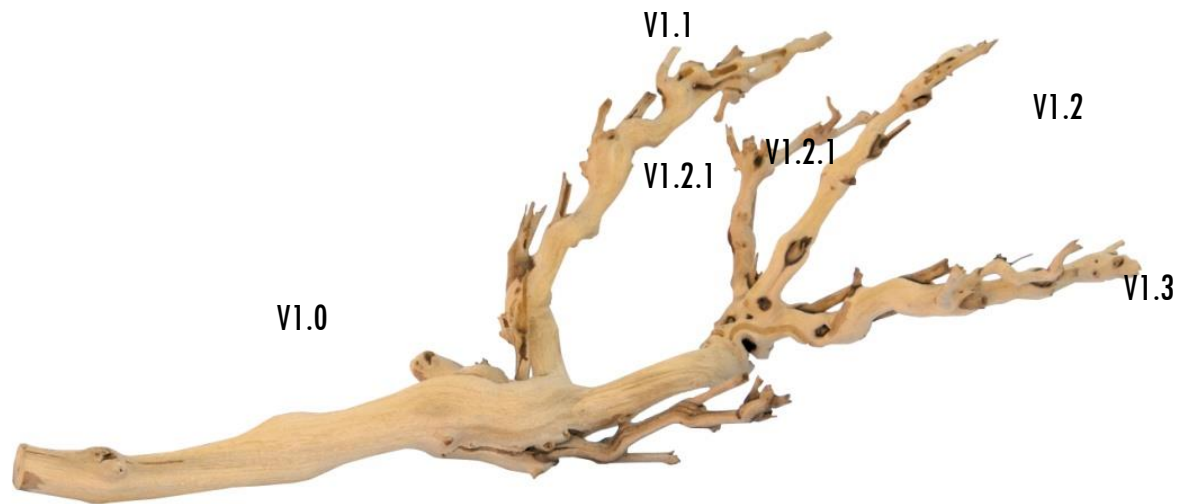
- 2 - Control de versiones
 - Repositorio
 - Se almacenan los archivos actualizados e históricos de cambio del proyecto.
 - Versión
 - Determina un conjunto de archivos
 - Master
 - Conjunto de archivos principales del proyecto
 - Abrir rama – branch
 - Bifurcación del máster para trabajar sobre dos ramas de forma independiente
 - Desplegar – check-out
 - Copia de trabajo local desde el repositorio.
 - Publicar - Commit
 - Una copia de los cambios hechos a una copia local es escrita o integrada sobre repositorio.
 - Conflicto
 - Problema entre las versiones de un mismo documento
 - Cambio – diff
 - Representa una modificación específica a un
 - Integración – Merge
 - Fusión entre dos ramas del proyecto
 - Actualización – sync o update
 - Integra los cambios que han sido hechos en el repositorio y las copias locales



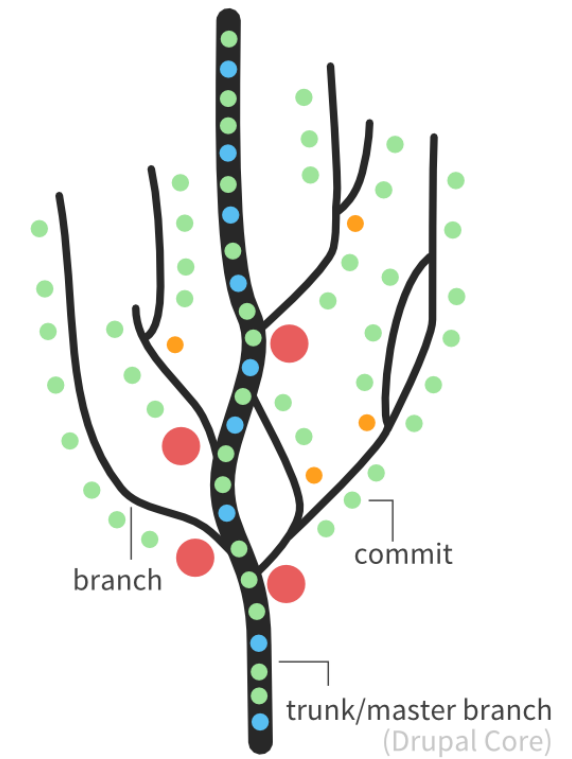
GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 2 - Control de versiones



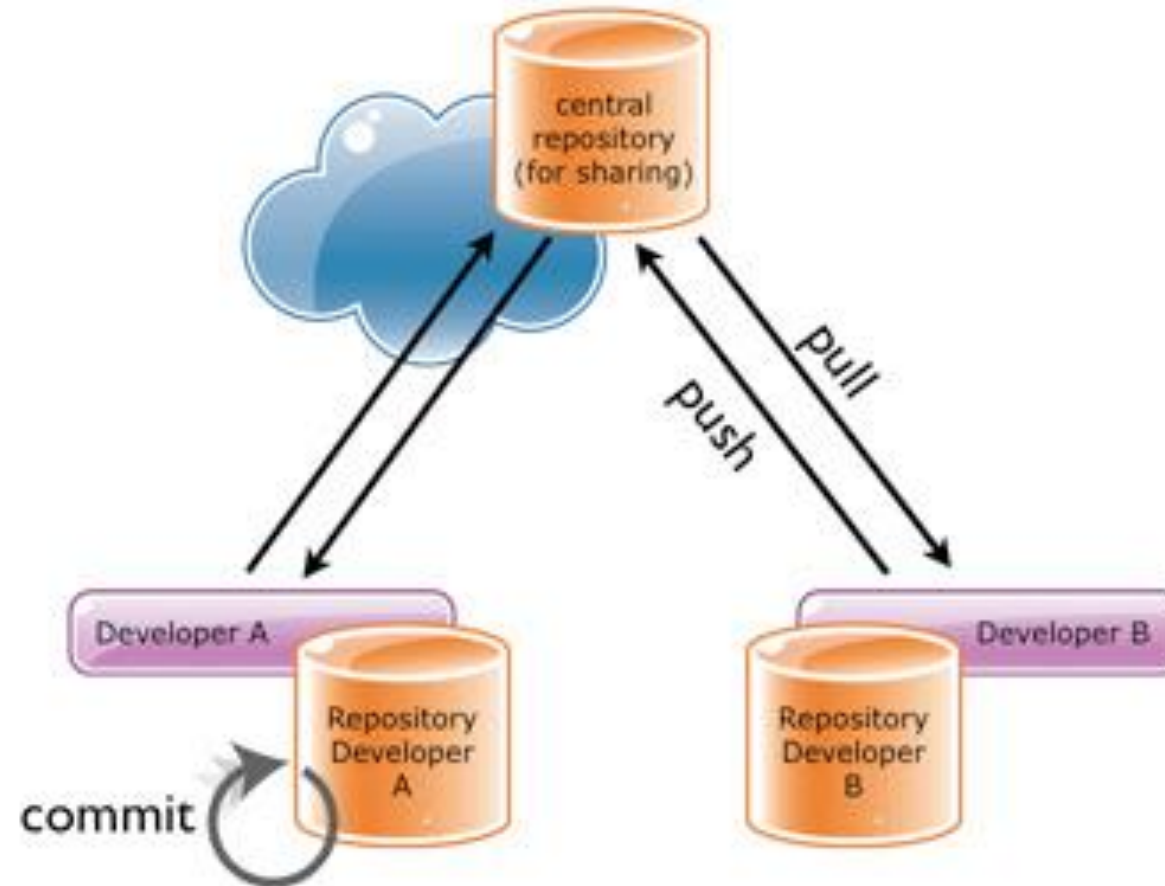
© Exo Terra - PT-3076



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

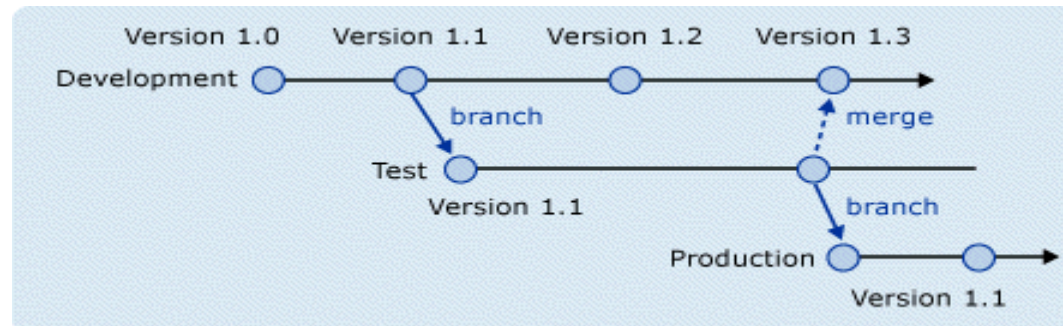
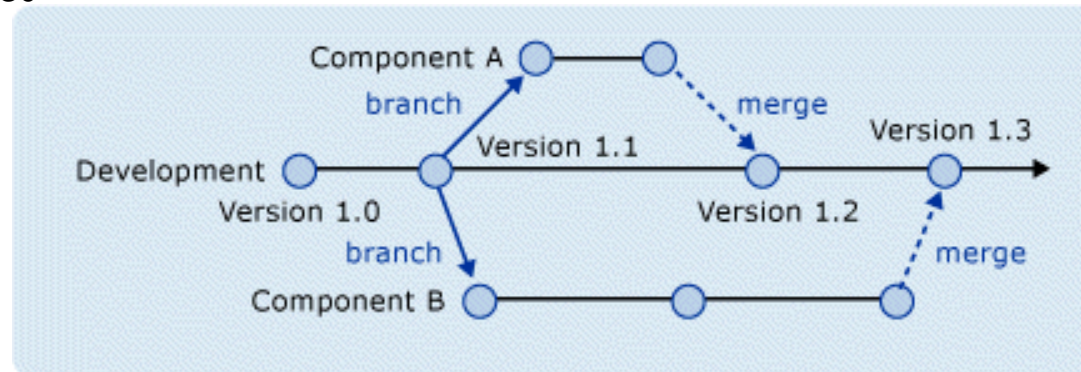
- 2 - Control de versiones



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

▪ 2 - Control de versiones



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

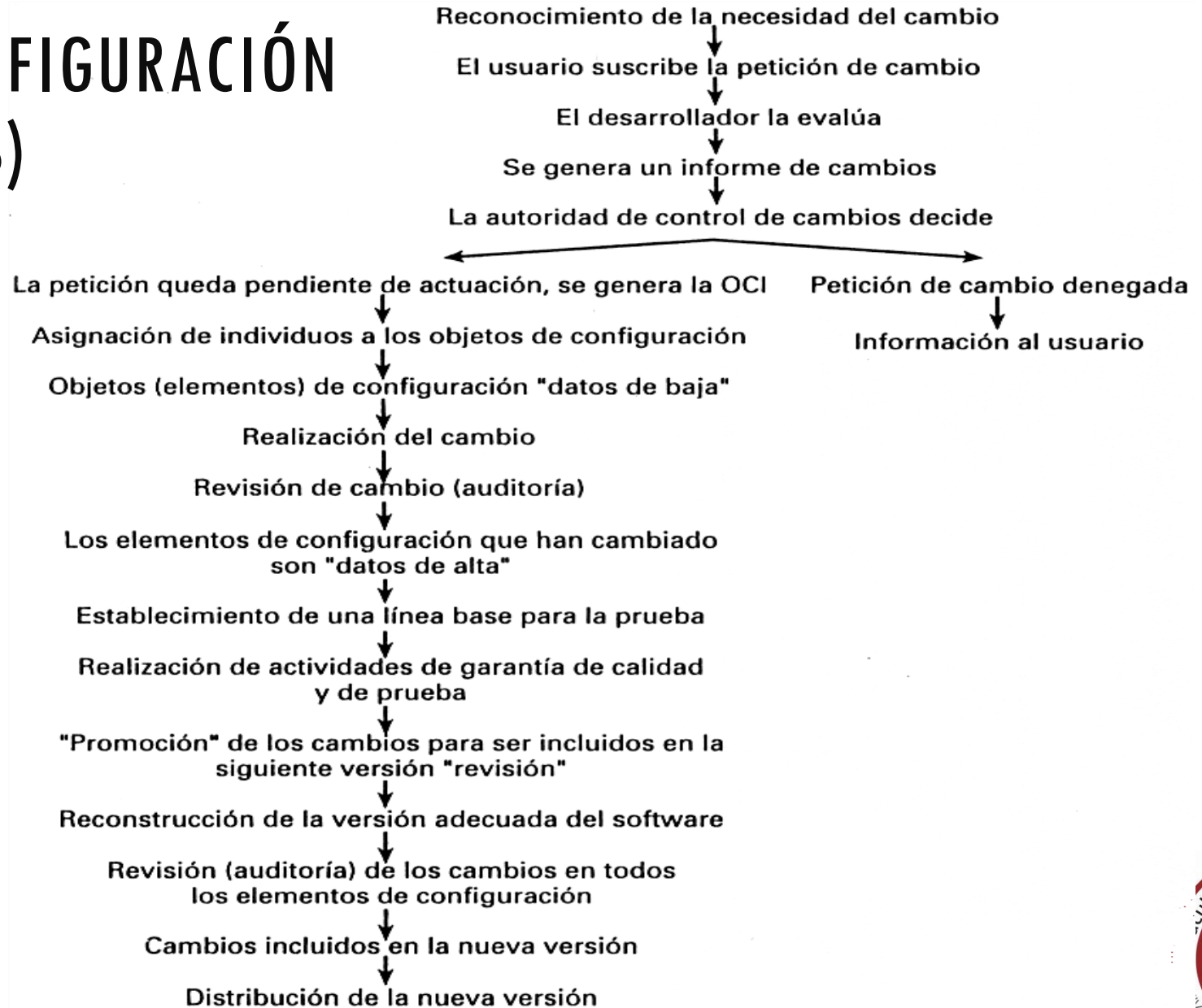
- 3 - Control de cambios
 - A lo largo del proyecto los cambios son inevitables y el control es vital para el desarrollo del mismo
 - Combina los procedimientos humanos y las herramientas adecuadas para proporcionar un mecanismo para el control del cambio



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 3 -Control de cambios



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 3 -Control de cambios
 - La autoridad de control de cambios (ACC) evalúa:
 - ¿Cómo impactará el cambio en el hardware?
 - ¿Cómo impactará el cambio en el rendimiento?
 - ¿Cómo alterará el cambio la percepción del cliente sobre el producto?
 - ¿Cómo afectará el cambio a la calidad y a la fiabilidad?
 - ...



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 4 - Auditoría de la configuración
 - La identificación y el control de versiones y el control de cambio, ayudan al equipo de desarrollo de software a mantener un orden, pero sólo se garantiza hasta que se ha generado la orden de cambio.
 - Cómo aseguramos que el cambio se ha realizado correctamente
 - Revisiones técnicas formales
 - Auditorías de configuración



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 4 - Auditoría de la configuración responde:
 - ¿Se ha hecho el cambio especificado en la Orden de Cambio?¿Se han incorporado modificaciones adicionales?
 - ¿Se ha llevado a cabo una RTF para evaluar la corrección técnica?
 - ¿Se han seguido adecuadamente los estándares de IS?
 - ¿Se han reflejado los cambios en el ECS: fecha, autor, atributos?
 - ¿Se han seguido procedimientos de GCS para señalar el cambio, registrarlo y divulgarlo?
 - ¿Se han actualizado adecuadamente todos los ECS relacionados?



GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE (GCS)

»Proceso de la GCS

- 5 - Generación de informes de estado de la configuración
 - Responde
 - ¿Qué pasó?
 - ¿Quién lo hizo?
 - ¿Cuándo pasó?
 - ¿Qué más se vio afectado?
- La generación de informes de estado de la configuración desempeña un papel vital en el éxito del proyecto



MÉTRICAS

ELEMENTOS CLAVE DE LA GESTIÓN DE PROYECTOS

Calendario temporal

Organización del personal

Análisis de riesgos

Seguimiento y control

Métricas

Estimaciones



Métricas y Estimaciones

MÉTRICAS

- » Las métricas son la clave tecnológica para el desarrollo y mantenimiento exitoso del software. (Briand et al., 1996)
- » En general, la medición persigue los siguientes objetivos fundamentales (Fenton y Pfleeger, 1997):
 - entender qué ocurre durante el desarrollo y el mantenimiento
 - controlar qué es lo que ocurre en nuestros proyectos
 - mejorar nuestros procesos y nuestros productos
 - Evaluar la calidad.



MÉTRICAS – DEFINICIONES

» Medida:

- indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.

» Medición:

- es el acto de determinar una medida.

» Métrica:

- medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. El ingeniero de software recopila medidas y desarrolla métricas para obtener indicadores.

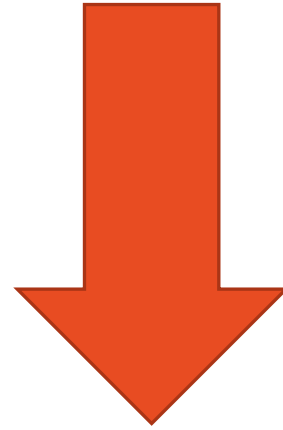
» Indicador:

- combinación de métricas. Proporciona una visión profunda que permite al gestor de proyectos o a los ingenieros de software ajustar el producto, el proyecto o el proceso para que las cosas salgan mejor



MÉTRICAS

» Las métricas pueden ser utilizadas para que los profesionales e investigadores puedan tomar las mejores decisiones



Métricas como medio para asegurar la calidad
en los Productos/Procesos/ Proyectos Software

MÉTRICAS

» Existen dos formas en que pueden usarse las mediciones de un sistema de software:

- Para asignar un valor a los atributos de calidad del software.
- Para identificar los componentes del sistema cuya calidad esta por debajo de un estándar.

» Métricas de control

- Apoyan la gestión del proceso.
 - Ej: esfuerzo promedio, tiempo requerido para reparar defectos

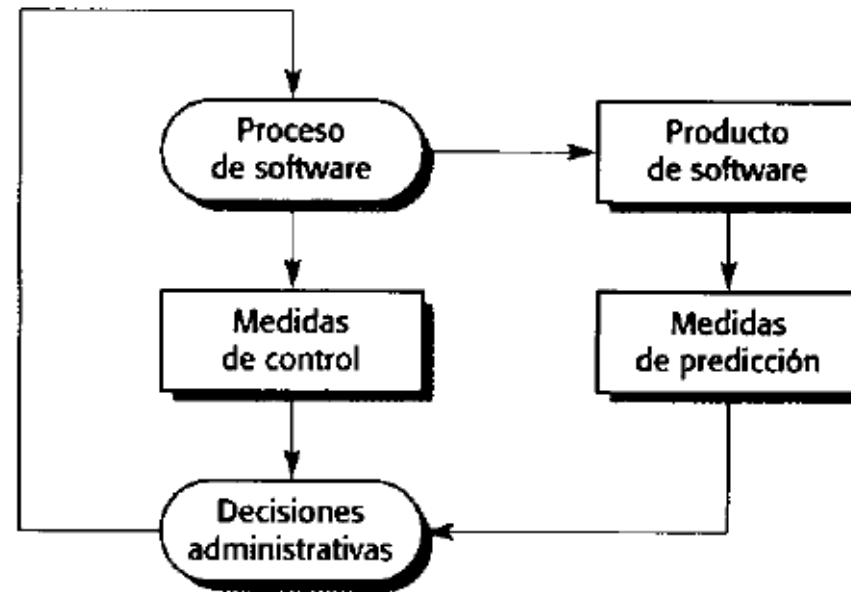
» Métricas de predicción (del producto)

- Ayudan a predecir las características del software. Se conocen también como métricas del producto.
 - Por ej.: Tamaño, complejidad



CLASIFICACIÓN DE LAS MÉTRICAS

» Tanto las métricas de control como las de predicción influyen en la toma de decisiones.



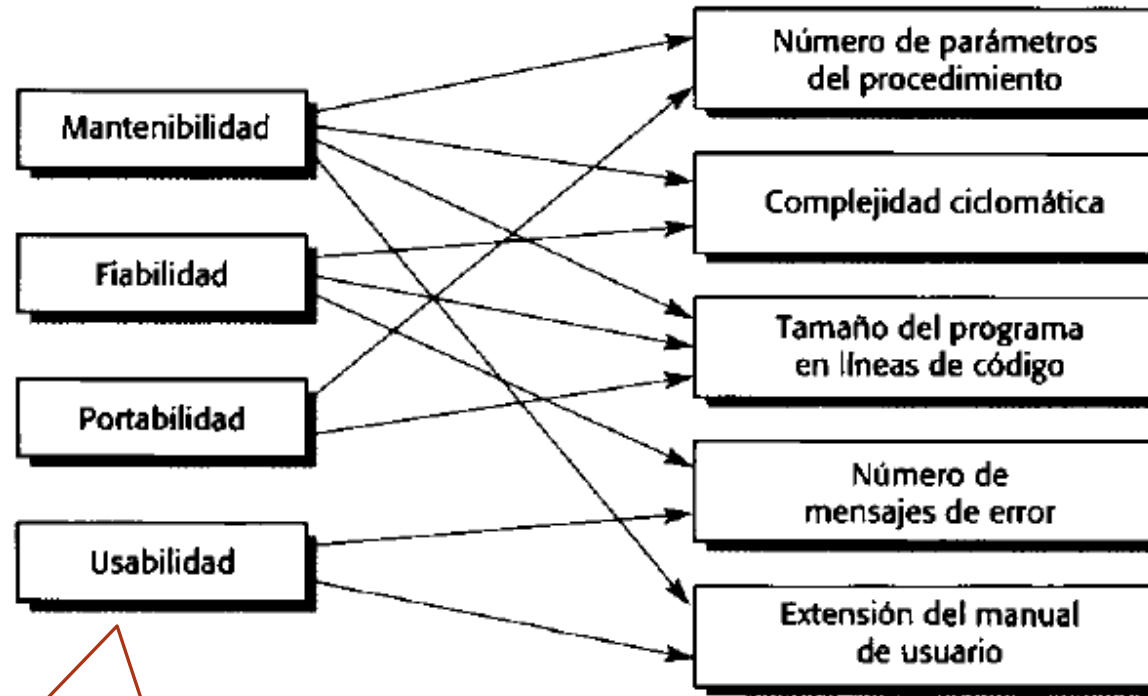
MÉTRICAS

- » Es difícil hacer mediciones directas de muchos de los atributos de calidad.
- » Los atributos externos se ven afectados por factores subjetivos como la experiencia, educación del usuario. Para evaluarlos hay que medir algunos atributos internos del software y relacionarlos.



MÉTRICAS

Atributos de calidad externos Atributos internos



Atributos que se refieren a como los desarrolladores y usuarios experimentan, el software

La relación entre atributos internos y externos debe comprenderse, validarse y expresarse en términos de una fórmula o modelo.

MÉTRICAS DEL PRODUCTO

» Las métricas del producto son métricas de predicción para medir los atributos internos de un sistema de software

» Se dividen en dos clases:

- Métricas dinámicas

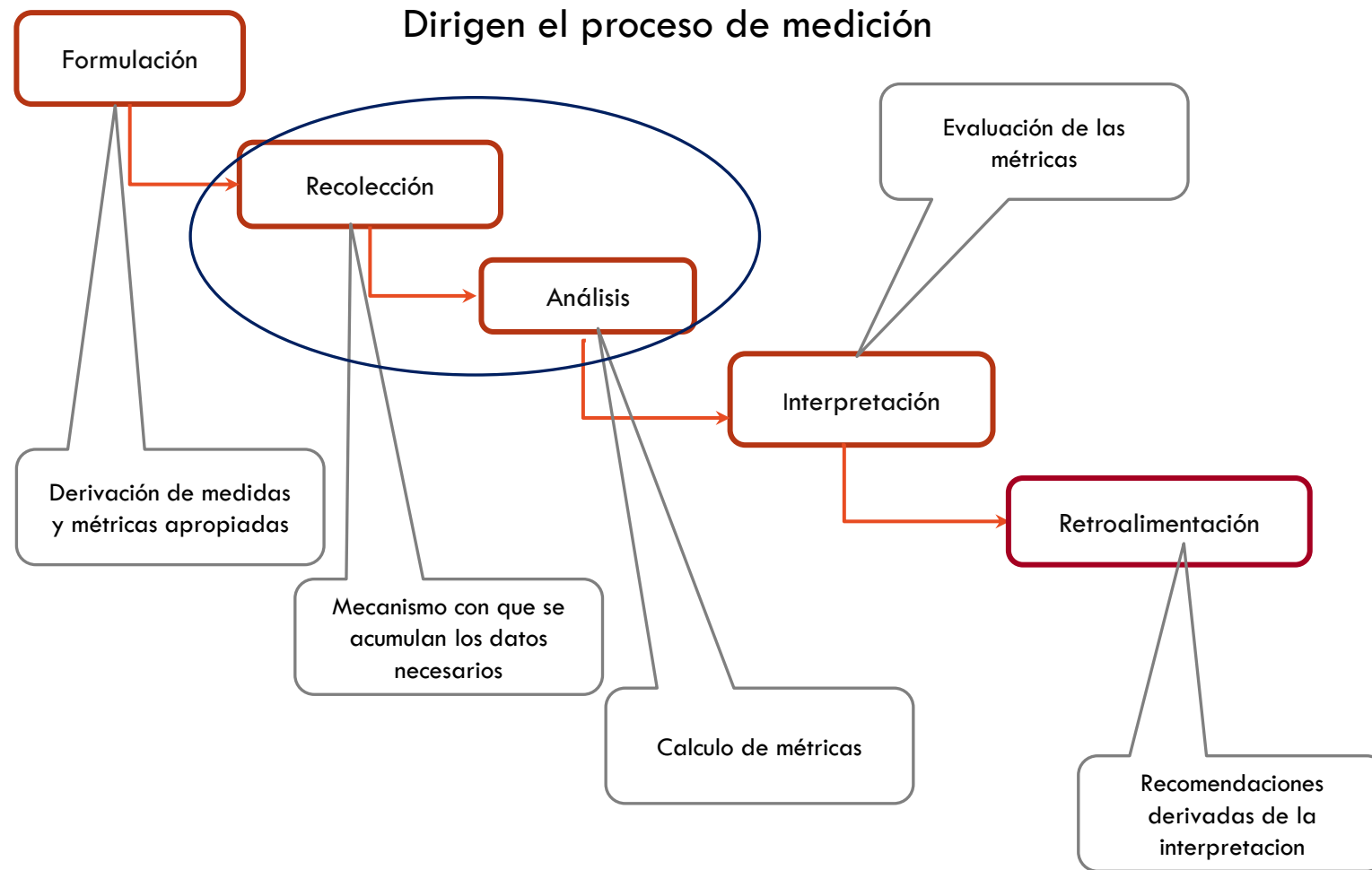
- Se recopilan de un programa en ejecución.
- Ayudan a valorar la eficiencia y fiabilidad de un programa.
- Ej. Nro de reportes de bugs.

- Métricas estáticas

- Que se recopilan mediante mediciones hechas de representaciones del sistema.
- Ayudan a valorar la complejidad, comprensibilidad y mantenibilidad de un sistema de software o sus componentes
- Ej: tamaño del código.



PROCESO DE MEDICIÓN



MÉTRICAS DEL PRODUCTO

- » Las métricas solo serán útiles si están caracterizadas de manera efectiva y se validan para probar su valor.
- » Principios que se pueden usar para caracterizar y validar las métricas
 - Una métrica debe tener propiedades matemáticas deseables (rango significativo)
 - Cuando una métrica representa una característica de software que aumenta cuando se presentan rasgos positivos o que disminuye al encontrar rasgos indeseables, el valor de la métrica debe aumentar o disminuir en el mismo sentido.
 - Cada métrica debe validarse empíricamente en una amplia variedad de contextos antes de publicarse o aplicarse en la toma de decisiones.



MÉTRICAS DEL PRODUCTO

» La métrica mas común para el tamaño de un producto es el numero de líneas de código.

» LDC - LÍNEAS DE CÓDIGO

- Medida directa del software y del proceso

Medida discutida porque depende del lenguaje y es post-mortem

- Es para saber en qué tiempo voy a terminar el software y cuántas personas voy a necesitar.
- Si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño
- Conformar una línea base para futuras métricas
- Ayudar al mantenimiento conociendo la complejidad lógica, tamaño, flujo de información, identificando módulos críticos
- Ayudar en los procesos de reingeniería



MÉTRICAS ORIENTADAS AL TAMAÑO

» Métricas derivadas del proceso de desarrollo:

- Productividad: relación entre KLDC y Persona mes
- Calidad: relación entre Errores y KLDC
- Costo: relación entre \$ y KLDC

Productividad = $\text{KLDC} / \text{persona-mes}$

Calidad = $\text{errores} / \text{KLDC}$

Documentación = $\text{págs.. Doc.} / \text{KLDC}$

Costo = $\text{\$/KLDC}$

KLDC (miles de líneas de código)



MÉTRICAS ORIENTADAS AL TAMAÑO

»LDC - LÍNEAS DE CÓDIGO

»Exigen explicar el manejo de:

- líneas en blanco, líneas de comentarios , declaraciones de datos, líneas con varias instrucciones separadas
- Información que se pierde: espacio que ocupa en disco, páginas que requiere el listado.

»Propuesta Fenton/Pfleeger

- Medir :
 - $CLOC = \text{Cantidad de líneas de comentarios}$
- Luego:
 - $\text{long total (LOC)} = NCLOC + CLOC$
- Surgen medidas indirectas:
 - $CLOC/LOC$ mide la densidad de comentarios



EJEMPLO

»Calcular, usando LDC , la productividad, calidad y costo para los cuatro proyectos de los cuales se proporcionan los datos.

Proyecto	LDC	U\$S	Error es	Personas-mes	Errores/KLDC	U\$S/KLDC	KLDC/P-Mes
P1	25.500	15000	567	15	22,23	588,23	1,7
P2	19.100	7200	210	10	10,99	376,96	1,91
P3	10.700	6000	100	20	9,34	560,74	0,53
P4	100.000	18000	2200	30	22	180	3,33

Productividad = $\text{KLDC} / \text{persona-mes}$

Calidad = $\text{errores} / \text{KLDC}$

Documentación = $\text{págs.. Doc.} / \text{KLDC}$

Costo = $\text{\$/KLDC}$

»¿Cuál es el proyecto de mayor calidad (errores/KLDC)?

»¿Cuál es el proyecto de mayor costo por línea (\$/KLDC)?

»¿Cuál es el proyecto de menor productividad por persona (KLDC/personas-mes)?



MÉTRICAS DE CONTROL (PREDICCIÓN)

»Un ejemplo de métrica de control es la llamada Métrica de Punto Función (FP), que examina el modelo de análisis con la intención de predecir el tamaño del sistema.

- Mide la cantidad de funcionalidad de un sistema descrito en una especificación



MÉTRICA DE PUNTO FUNCIÓN

PF- Punto función (Albrecht 1978)

Factor de Ponderación, es subjetivo y esta dado por la organización/equipo

$$PF = TOTAL * [0.65 + 0.01 * SUM(F_i)] \quad i=1 \text{ a } 14 \quad 0 \leq F_i \leq 5$$

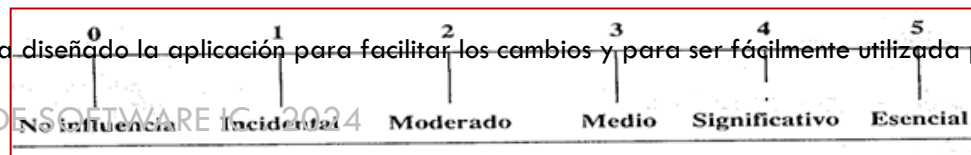
	simple	medio	complejo	
# Entradas	* [3 4 6]	=	
# Salidas	* [4 5 7]	=	
# Consultas	* [3 4 6]	=	
# Almacenamientos internos	* [7 10 15]	=	
# Interfaces externas	* [5 7 10]	=	
				TOTAL

Son valores de ajuste de la complejidad según las preguntas de la siguiente pantalla

Medida subjetiva independiente del lenguaje, de estimación más fácil. Métrica temprana

MÉTRICA DE PUNTO FUNCIÓN

- » 1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
- » 2. ¿Se requiere comunicación de datos?
- » 3. Existen funciones de procesamiento distribuido?
- » 4. ¿Es crítico el rendimiento?
- » 5. ¿Se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?
- » 6. ¿Requiere el sistema entrada de datos interactiva?
- » 7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
- » 8. ¿Se actualizan los archivos maestros de forma interactiva?
- » 9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
- » 10. ¿Es complejo el procesamiento interno?
- » 11. ¿Se ha diseñado el código para ser reutilizable?
- » 12. ¿Están incluidas en el diseño la conversión y la instalación'?
- » 13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
- » 14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?



Cada una de las preguntas se contesta de acuerdo a la siguiente escala de valores



MÉTRICA DE PUNTO FUNCIÓN

» Métricas derivadas:

- Productividad: relación entre PF y Persona_mes
- Calidad: relación entre Errores y PF
- Costo: relación entre \$ y PF

$$\begin{aligned}\textbf{Productividad} &= \text{PF} / \text{Persona_mes} \\ \textbf{Calidad} &= \text{Errores} / \text{PF} \\ \textbf{Costo} &= \$ / \text{PF}\end{aligned}$$



EL DESARROLLO DE UNA MÉTRICA

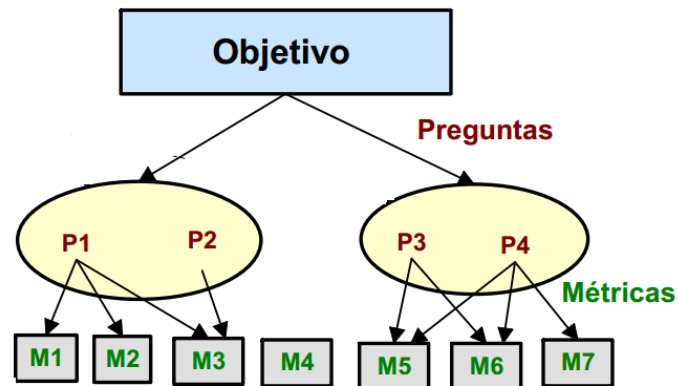
- » Victor Basili desarrolló un método llamado GQM (Goal, Question, Metric) (o en castellano: OPM Objetivo, Pregunta, Métrica).
- » (<ftp://ftp.cs.umd.edu/pub/sel/papers/gqm.pdf>)
- » Dicho método esta orientado a lograr una métrica que “mida” cierto objetivo . El mismo nos permite mejorar la calidad de nuestro proyecto.
- » Es útil para decidir qué medir.
- » Debe estar orientado a metas.
- » Es flexible.



GQM (OPM)

»Estructura :

- Nivel Conceptual (Goal / Objetivo).
 - Se define un objetivo (en nuestro caso, para el proyecto).
- Nivel Operativo (Question / Pregunta).
 - Se refina un conjunto de preguntas a partir del objetivo, con el propósito de verificar su cumplimiento.
- Nivel Cuantitativo (Metric / Métrica).
 - Se asocia un conjunto de medidas para cada pregunta, de modo de responder a cada una de un modo cuantitativo.



GQM EJEMPLO

»Evaluamos , en la etapa de Análisis de Requerimientos, la tarea Asignación de responsabilidades (es sólo un ejemplo, se puede tomar la actividad o tarea que se crea prioritaria).

Propósito	Evaluar	
Característica	Asignación de responsabilidades	
Punto de Vista	Gerencia de Proyecto	
Pregunta 1	¿Existe un proceso para la asignación de roles?	
	M1	Valor Binario
Pregunta 2	¿Hay un responsable de asignar roles?	
	M2	Valor Binario
Pregunta 3	¿El responsable siempre realiza su tarea?	
	M3	Valor Binario
Pregunta 4	¿Existe información anterior sobre las tareas realizadas por cada integrante?	
	M4	Valor Binario
Pregunta 5	¿Esa información esta disponible?	
	M5	Valor Binario



GQM EJEMPLO

»Indicadores

<u>Nombre</u>	<u>Descripción</u>	<u>Fórmula</u>
I1	Gestión de Asignación de roles	M2 & M3 & M4 & M5
I2	Proceso de Asignación de roles	M1 & M4 & M5

A partir de los indicadores definidos, se propone realizar el control de la meta a través de un tablero de control de indicadores específicos. Podemos decir que nuestra meta se cumple si los indicadores muestran los siguientes valores:

I1	Gestión de Asignación de roles	Verdadero
I2	Proceso de Asignación de roles	Verdadero



ESTIMACIONES

ESTIMACIONES

» Estimación

- Técnicas que permiten dar un valor aproximado.
- Para obtener estimaciones confiables generalmente se usan varias técnicas y se comparan y concilian resultados.
- La estimación no es una ciencia exacta.
- Modificaciones en la especificación hacen peligrar las estimaciones.
- Requiere experiencia, acceso a información histórica y decisión para convertir información cualitativa en cuantitativa.



ESTIMACIONES

El riesgo de la estimación decrece con la disponibilidad de historia

- Se realizan estimaciones de recursos, costos y tiempos
- Los factores que influyen son la complejidad, el tamaño, la estructuración del proyecto.



ESTIMACIONES DE COSTOS

- » Existen tres principales parámetros que se deben usar al calcular los costos de un proyecto.
- » Costos de esfuerzo (pagar desarrolladores e ingenieros)
- » Costos de hardware y software, incluido el mantenimiento
- » Costos de viaje
- » Para la mayoría de los proyectos, el mayor costo es el primer rubro.



ESTIMACIONES DE COSTOS

- » Debe estimarse el esfuerzo total (meses-hombre), sin embargo se cuenta con datos limitados para esta valoración.
- » Es posible que se deba licenciar el middleware y la plataforma, o que se requieran mayor cantidad de viajes cuando se desarrolla en distintos lugares.
- » Se debe iniciar con un bosquejo de Plan de Proyecto y se debe contar con una especificación de los requerimientos.



FIJACIÓN DE PRECIO

- »En principio el precio es simplemente el costo de desarrollo, sin embargo en la practica , la relación entre el costo y el precio al cliente no es tan simple.
- »Cuando se calcula un precio hay que considerar temas de índole organizacional, económica, política y empresarial.



ESTIMACIONES DE RECURSOS

- » Recursos humanos
- » Recursos de software reutilizables
- » Recursos de hard y herramientas de software

- » Cada recurso requiere:
 - Descripción
 - Informe de disponibilidad
 - Fecha en que se lo requiere
 - Tiempo que se lo necesita



TÉCNICAS DE ESTIMACIÓN

» Juicio experto:

- Se consultan varios expertos. Cada uno de ellos estima. Se comparan y discuten

» Técnica Delphi:

- Consiste en la selección de un grupo de expertos a los que se les pregunta su opinión. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, con el objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes.

» División de trabajo:

- Jerárquica hacia arriba



MODELOS EMPÍRICOS DE ESTIMACIÓN

- » Utilizan fórmulas derivadas empíricamente para predecir costos o esfuerzo requerido en el desarrollo del proyecto.
- » Ej: MODELO COCOMO de Boehm (1981)
- » (COConstructive COst MOdel, modelo constructivo de costos) se obtuvo recopilando datos de varios proyectos grandes.
- » Las formulas que utiliza el COCOMO vinculan el tamaño del sistema y del producto, factores del proyecto y del equipo con el esfuerzo necesario para desarrollar el sistema.



ESTIMACIONES COCOMO 81

» Modelos empíricos de estimación

El modelo inicial consideraba tres tipos de proyectos:

Orgánicos: Proyectos pequeños y de poca gente

Semiacoplados: Proyectos intermedios

Empotrados: Proyectos con restricciones rígidas

Modelo COCOMO

El tipo BASICO estima a través de LDC

Tipos de proyectos: Orgánicos - Semiacoplados - Empotrados

a	2.4	3.0	3.6
b	1.05	1.12	1.20

$$E = a (KLDC)^b \quad \text{Esfuerzo en persona-mes}$$

Fórmula básica de nivel 1 de estimación, había tres niveles y profundizaban el detalle de la estimación.

También suponía un modelo de proceso en cascada con uso de lenguajes imperativos del estilo "C" o fortran

ESTIMACIONES COCOMO 81

Modelo COCOMO

Tipos de proyectos: Orgánicos - Semiacoplados - Empotrados

c	2.5	2.5	2.5
d	0.38	0.35	0.32

$D = cE^d$ Duración en meses

Formula de nivel 1
para el calculo de
duración en meses

»Ejemplo:

»Suponer que se cuenta con un Proyecto con: LDC : 19.100 y Semiacoplado

» $E = 3 * (19,1)1,12 = 81,63$ esfuerzo en personas/mes

» $D = 2.5 * E^{0.35} = 11.67$ meses

El esfuerzo se expresa en meses/persona (PM) y

representa los meses de trabajo de una

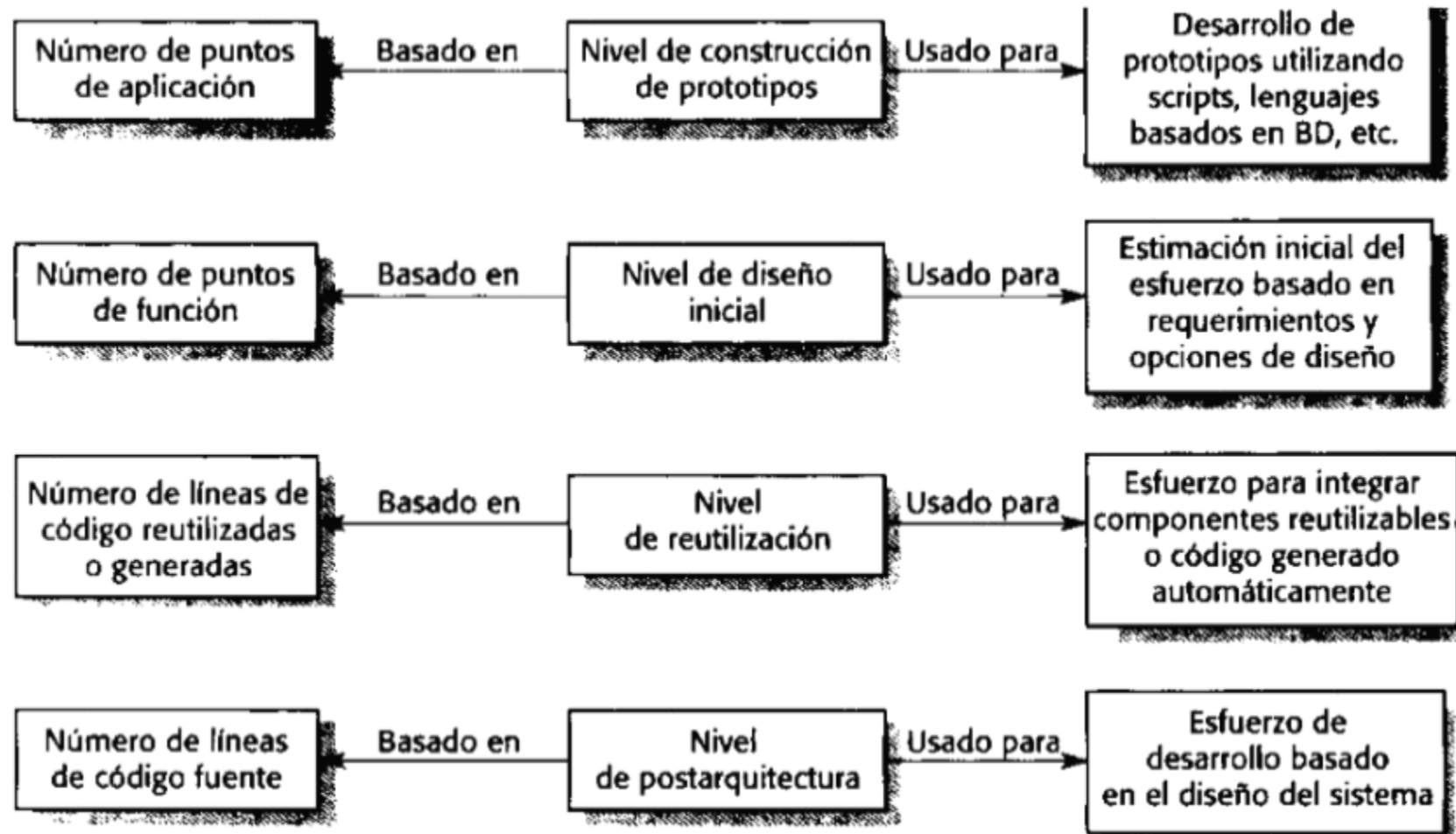
persona fulltime, requeridos para desarrollar el proyecto

COCOMO II (2000)

- » Reconoce que las líneas de código son difíciles de estimar tempranamente. Considera diferentes enfoques para el desarrollo , como la construcción de prototipos, el desarrollo basado en componentes, desarrollo en espiral y engloba varios niveles que producen estimaciones detalladas de forma incremental.
- » COCOMO II está compuesto por 4 niveles:
 - De construcción de prototipos
 - De diseño inicial
 - De reutilización
 - De post-arquitectura



NIVELES DE COCOMO II



COCOMO II 1. NIVEL CONSTRUCCIÓN DE PROTOTIP

Punto de aplicación = Punto objeto

»La fórmula para el cálculo del esfuerzo para el prototipo del sistema es:

$$»PM = (NAP \times (1 - \%reutilización/100))/PROD$$

»PM = esfuerzo estimado en personas/mes

»NAP = total de puntos de aplicación

»PROD = productividad

1. El número de pantallas independientes que se despliegan. Las pantallas sencillas cuentan como 1 punto objeto, las pantallas moderadamente complejas cuentan como 2 y las pantallas muy complejas cuentan como 3 puntos objeto.
2. El número de informes que se producen. Los informes simples cuentan como 2 puntos objeto, los informes moderadamente complejos cuentan como 5, y los informes que son difíciles de producir cuentan como 8 puntos objeto.
3. El número de módulos en lenguajes imperativos como Java o C++ que deben ser desarrollados para complementar el código de programación de la base de datos se contabilizará como 10 puntos objeto.

COCOMO II

1. NIVEL DE CONSTRUCCIÓN DE PROTOTIPOS

»Productividad de puntos de objeto, se basa en la siguiente tabla para obtener el PROD de la formula anterior

Experiencia y capacidad de los desarrolladores	Muy baja	Baja	Media	Alta	Muy alta
Madurez y capacidad de las herramientas CASE	Muy baja	Baja	Media	Alta	Muy alta
PROD (NOP/mes)	4	7	13	25	50



COCOMO II 2. NIVEL DE DISEÑO INICIAL

»La fórmula para las estimaciones en este nivel es:

»Esfuerzo = $A \times \text{Tamaño} \times B \times M$.

»Boehm propone que $A = 2.94$. (Otros autores proponen: 2.45)

»Tamaño = KLDC (miles de líneas de código fuente).

» $B = 0.91 \times \sum SF_j$ (ver tabla)

» $M = \text{PERS} \times \text{RCPX} \times \text{RUSE} \times \text{PDIF} \times \text{PREX} \times \text{FCIL} \times \text{SCED}$ (ver mas adelante)



COCOMO II - 2. NIVEL DE DISEÑO INICIAL TABLA DE VALORES DE ESCALA SF PARA CALCULAR B

	Muy bajo	bajo	nominal	alto	Muy alto	Extra alto
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

PREC: experiencia de proyectos precedentes. Desde totalmente sin precedentes hasta totalmente familiar

FLEX: flexibilidad de desarrollo. Desde requerimientos muy rígidos hasta solamente metas generales

RESL: Nivel de riesgos y tiempo dedicado a arquitectura, desde casi nada a total **TEAM:** cohesión del equipo. Desde dificultades graves en interacción hasta interacción suave.

PMAT: Madurez de acuerdo a CMM: Desde 1 hasta 5 (uno se repite)



COCOMO II — 2. NIVEL DE DISEÑO INICIAL- ACLARACIÓN SIGLAS DE M

- »M: Multiplicador.
- »Son siete características/atributos del proyecto y del proceso que influyen en la estimación. Éstas hacen que aumente o disminuya el esfuerzo requerido.
- »• RCPX = Fiabilidad y complejidad del producto.
- »• RUSE = Reutilización requerida.
- »• PDIF = Dificultad de la plataforma.
- »• PERS = Capacidad del personal.
- »• PREX = Experiencia del personal.
- »• SCED = Calendario.
- »• FCIL = Facilidades de apoyo.
- »Se pueden estimar directamente en una escala de 1 (valor muy bajo) a 6 (valor muy alto).



COCOMO II - 3. NIVEL DE REUTILIZACIÓN

» Para el código generado automáticamente, el modelo estima el número de persona/mes necesarias para integrar este código.

» $PM_{auto} = (ASLOC \times AT / 100) / ATPROD$.

» AT = porcentaje de código adaptado que se genera automáticamente.

» ATPROD = productividad de los ingenieros que integran el código

» ASLOC = Nro de líneas de código en los componentes que deben ser adaptadas



COCOMO II - 4. NIVEL DE POST-ARQUITECTURA

- » Las estimaciones están basadas en la misma fórmula básica
- » $PM = A \times \text{Tamaño}B \times M$
- » pero se utiliza un conjunto más extenso de atributos (17 en lugar de 7) de producto, proceso y organización para refinar el cálculo del esfuerzo inicial.
- » La estimación del número de líneas de código se calcula utilizando tres componentes:
 - Una estimación del número total de líneas nuevas de código a desarrollar.
 - Una estimación del número de líneas de código fuente equivalentes (ESLOC) calculadas usando el nivel de reutilización.
 - Una estimación del número de líneas de código que tienen que modificarse debido a cambios en los requerimientos.
- » Estas estimaciones se añaden para obtener el tamaño del código (KLDC).



COCOMO II - 4. NIVEL DE POST-ARQUITECTURA

- »Estas estimaciones se añaden para obtener el tamaño del código (KLDC).
- »El exponente B se calcula considerando 5 factores de escala.
- »Como
 - Productividad de desarrollo
 - Cohesión del equipo
 - Entre otros



ESTIMACIONES ON-LINE

»Visitar las páginas:

»Para COCOMO81:

»http://sunset.usc.edu/research/COCOMOII/cocomo81_pgm/cocomo81.html

»Para COCOMO 2:

»<http://csse.usc.edu/tools/COCOMOII.php>



CALIDAD DE SOFTWARE

INTRODUCCIÓN

» ¿Qué es la calidad?

Satisfacción...



1. DEFINICIÓN DE CALIDAD

»calidad

-(Del latín)

1.f. Pro
es de k

2.f. Bu
merca

3.f. Carácter, genio, índole.

4.f. Condición o requisito que se pone en un contrato.

5.f. Estado de una persona, naturaleza, edad y demás circunstancias y condiciones que se requieren para un cargo o dignidad.

Se ve una serie de definiciones relacionadas, la mas destacable es la primera donde se habla de “**propiedades que pueden ser juzgadas**” de ahí se desprende que la calidad es un termino totalmente subjetivo, que va a depender del juicio de la persona que intervenga en la evaluación.

“Esta tela

do los



1. DEFINICIÓN DE CALIDAD

»A lo largo de la historia se han desarrollado filosofías o culturas de calidad, de las cuales algunas han sobresalido porque han tenido resultado satisfactorios.

»A los que realizaron estas filosofías se los ha llamado **Padres de la Calidad**.



¿QUE ES LA CALIDAD?

- » Calidad es un concepto manejado con bastante frecuencia en la actualidad, pero a su vez, su significado es percibido de distintas maneras.
- » Al hablar de bienes y/o servicios de calidad, la gente se refiere normalmente a bienes de lujo o excelentes con precios elevados.
- » Su significado sigue siendo ambiguo y muchas veces su uso depende de lo que cada uno entiende por calidad, por lo cual es importante comenzar a unificar su definición.



¿QUE ES LA CALIDAD?

»Reconocimiento del mercado - Autos



QUE ES LA CALIDAD?

¿Cuál fue el celular más vendido en el 2023?



Apple iPhone 14 Pro Max: 26,5 millones.
Apple iPhone 14 Pro: 21 millones.
Apple iPhone 14: 16,5 millones.
Apple iPhone 13: 15,5 millones.

Samsung Galaxy A14: 12,4 millones.
Samsung Galaxy S23 Ultra: 9,6 millones.
Samsung Galaxy A14 5G: 9 millones.
Samsung Galaxy A54 5G :8,8 millones.
Samsung Galaxy A34 5G: 7,1 millones.
Apple iPhone 11: 6,9 millones.

Los dispositivos que Apple logró posicionar en el ranking de Counterpoint Research

¿QUE ES LA CALIDAD?



¿Cual elegís?

¿QUÉ ES LA CALIDAD ?

- » Las principales normas internacionales definen la calidad como :
- » “El grado en el que un conjunto de características inherentes cumple con los requisitos” (ISO 9000)
- » “Conjunto de propiedades o características de un producto o servicio que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas” (ISO 8402)



CALIDAD DE LOS SISTEMAS DE INFORMACIÓN

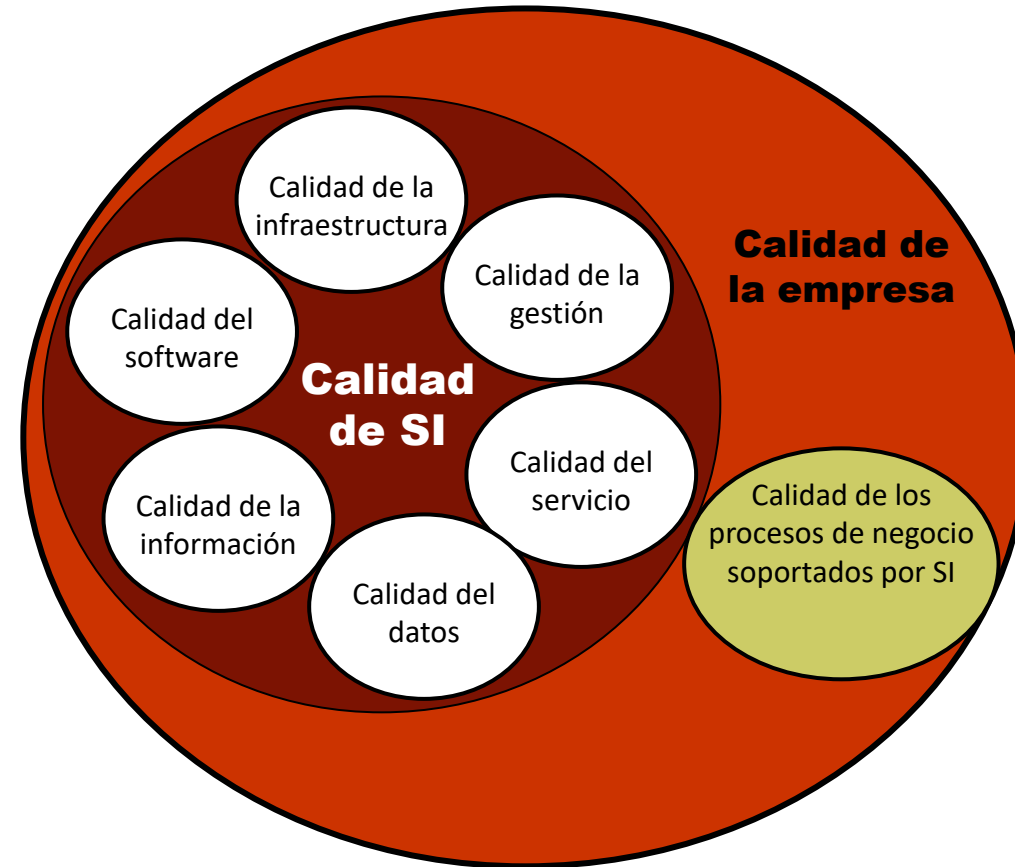
- » La importancia de los sistemas de información (SI) en la actualidad hace necesario que las empresas de tecnología hagan mucho hincapié en los estándares de calidad.
- » Stylianou y Kumar plantean que se debe apreciar la calidad desde un todo, donde cada parte que la componen debe tener su análisis de calidad.



COMPONENTES

Visión holística de la calidad

Stylianou y Kumar
(2000)



COMPONENTES

» Calidad de la Infraestructura

- incluye, por ejemplo, la calidad de las redes, y sistemas de software.

» Calidad de Software

- de las aplicaciones de software construidas, o mantenidas, o con el apoyo de IS.

» Calidad de Datos

- Que ingresan en el sistema de información.

» Calidad de Información

- está relacionada con la calidad de los datos.

» Calidad de gestión

- incluye el presupuestó , planificación y programación.

» Calidad de servicio

- incluye los procesos de atención al cliente

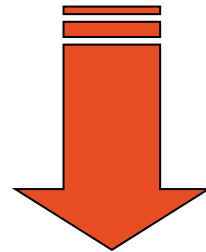


CALIDAD DE SOFTWARE

»La calidad del software se ha mejorado significativamente en estos últimos años, en particular por una mayor conciencia de la importancia de la gestión de la calidad y la adopción de técnicas de gestión de la calidad para desarrollo en la industria del software

»Se divide en

- Calidad del producto obtenido
- Calidad del proceso de desarrollo



Son dependientes

CALIDAD DEL PRODUCTO Y PROCESO

»Producto

- Un producto es un bien tangible que es el resultado de un proceso.
- Aunque el software tiene aspectos intangibles, un producto software es sin embargo un bien en sí mismo e incluye sus documentos asociados.
- La estandarización del producto define las propiedades que debe satisfacer el producto software resultante.

»Proceso

- La estandarización del proceso define la manera de desarrollar el producto software.

Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.



CLASIFICACIÓN DE NOI S Y MODELOS DE CALIDAD

Producto Software

Producto Software



Proceso de Desarrollo 

Proceso de Desarrollo 

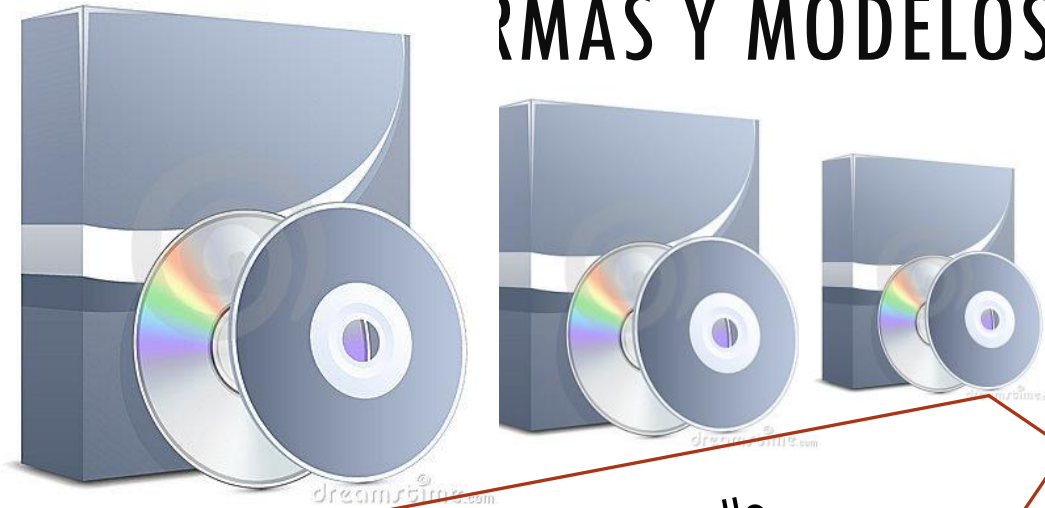
Organización 

CLASIFICACIÓN DE NORMAS Y MODELOS DE CALIDAD



CLASIF

RMAS Y MODELOS DE CALIDAD



PMBOOK - SWEBOOK- SIX SIGMA
ISO/IEC 12207 - ISO/IEC 15504 – ISO/IEC 90003
CMMI – SCAMPI – IDEAL - ISO/IEC 29110

CLASIFICACIÓN DE NOI

Producto Software



Proceso de Desarrollo

MODELOS DE CALIDAD

Producto Software



Proceso de Desarrollo



Organización

CALIDAD TOTAL – TQM – ISO/IEC 9001
SEGURIDAD DE LA INFORMACIÓN – ISO/IEC 27001



CLASIFICACIÓN DE NORMAS Y MODELOS DE CALIDAD

CALIDAD DE PRODUCTO DE SOFTWARE
CALIDAD DE USO – CALIDAD DE DATOS
ISO/IEC 9126 /14598 - ISO/IEC 25000

CALIDAD DE SERVICIOS
ISO/IEC 20000 - ITIL



PMBOOK - SWEBOOK- SIX SIGMA - ISO/IEC 12207 - ISO/IEC 15504 –
ISO/IEC 33000 - ISO/IEC 90003 -CMMI – SCAMPI – IDEAL - ISO/IEC 29110



CALIDAD TOTAL – TQM – ISO/IEC 9001
SEGURIDAD DE LA INFORMACIÓN – ISO/IEC 27001

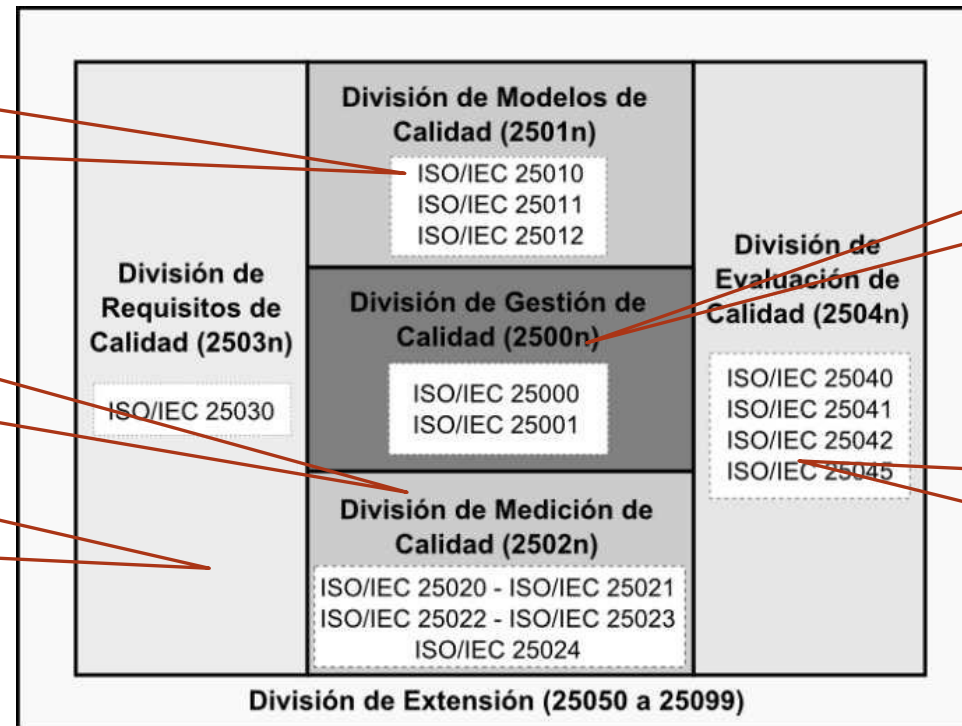
MODELO DE CALIDAD SQUARE ISO/IEC 25000

»ISO/IEC 25000 SQuaRE Software product Quality Requirement and Evaluation

Modelo de calidad detallado incluyendo características para calidad interna y externa y la calidad de datos.

Modelo de referencia de la medición de la calidad del producto, definiciones de medidas de calidad y guías prácticas de uso

Ayuda a especificar los requisitos de calidad que pueden ser usados en el proceso de elicitación.



Las normas que forman este apartado definen todos los modelos, términos y definiciones comunes referenciados por toda la serie SQuaRE

Requisitos, recomendaciones y guías para la evaluación de producto.



MODELO DE CALIDAD SQUARE ISO/IEC 25000

» ISO/IEC 2500n – División gestión de la calidad

ISO/IEC 25000:2005 - Guide to SQuaRE:

ISO/IEC 25001:2007 - Planning and Management.

» ISO/IEC 2501n – División modelos de calidad

ISO/IEC 25010 - System and software quality models

ISO/IEC 25012 - Data Quality model

» ISO/IEC 2502n – División de medición de calidad

ISO/IEC 25020 - Measurement reference model and guide

ISO/IEC 25021 - Quality measure elements

ISO/IEC 25022 - Measurement of quality in use

ISO/IEC 25023 - Measurement of system and software product quality.

ISO/IEC 25024 - Measurement of data quality

» ISO/IEC 2503n – División Requerimientos de calidad

ISO/IEC 25030 - Quality requirements

» ISO/IEC 2504n – División Evaluación de la calidad

90

ISO/IEC 25040 - Evaluation reference model and guide

ISO/IEC 25041 - Evaluation guide for developers, acquirers and independent evaluators

ISO/IEC 25042 - Evaluation modules.

ISO/IEC 25045 - Evaluation module for recoverability



MODELO DE CALIDAD SQUARE ISO/IEC 25010



MODELO DE CALIDAD SQUARE ISO/IEC 25010



MODELO DE CALIDAD SQUARE ISO/IEC 25000

»Ejemplo de métricas de Interoperabilidad

Table 8.1.3 Interoperability metrics

External interoperability metrics									
Metric name	Purpose of the metrics	Method of application	Measurement, formula and data element computations	Interpretation of measured value	Metric scale type	Measure type	Input to measurement	ISO/IEC 12207 SLCP Reference	Target audience
Data exchangeability (Data format based)	How correctly have the exchange interface functions for specified data transfer been implemented?	Test each downstream interface function output record format of the system according to the data fields specifications. Count the number of data formats that are approved to be exchanged with other software or system during testing on data exchanges in comparing with the total number.	$X = A / B$ A= Number of data formats which are approved to be exchanged successfully with other software or system during testing on data exchanges B= Total number of data formats to be exchanged	$0 \leq X \leq 1$ The closer to 1.0 is the better.	Absolute	A= Count B= Count X= Count/Count	Req. spec. (User manual) Test report	6.5 Validation	Developer
FOOTNOTE <i>It is recommended to test specified data transaction.</i>									
Data exchangeability (User's success attempt based)	How often does the end user fail to exchange data between target software and other software?	Count the number of cases that interface functions were used and failed.	a) $X = 1 - A / B$ A= Number of cases in which user failed to exchange data with other software or systems B= Number of cases in which user attempted to exchange data	$0 \leq X \leq 1$ The closer to 1.0 is the better.	a) Absolute	A= Count B= Count X= Count/Count	Req. spec. (User manual) Test report	5.4 Operation	Maintainer
	How often are the data transfers between target software and other software successful?		b) $Y = A / T$ T= Period of operation time	$0 \leq Y$ The closer to 0, is the better.	b) Ratio	Y= Count/Time T= Time			
	Can user usually succeed in exchanging data?								

MODELO DE CALIDAD DE DATOS - ISO/IEC 25012

Datos → Información → Conocimiento

- Si los datos no tienen suficiente calidad, entonces, se pueden convertir 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

MODELO DE CALIDAD DE DATOS - ISO/IEC 25012

- » Necesidad de una visión coherente e integrada de los datos para garantizar la interoperabilidad de los sistemas
- » La dispersión y la reproducción de estos datos entre diferentes organizaciones
- » La necesidad de reducir la ambigüedad semántica entre entidades en bases de datos: la misma definición se utiliza para diferentes fenómenos, o lo contrario
- » La frecuencia de intercambio de datos en internet, en algunos casos sin saber la calidad del proceso de producción de los mismos
- » La necesidad de realizar comparaciones internacionales
- » La necesidad de cumplir con leyes internacionales o reglamentaciones
- » La necesidad de reducir los costos por falta de calidad de los datos

MODELO DE CALIDAD DE DATOS - ISO/IEC 25012

» La norma entiende por calidad de datos:

- La capacidad de las características de los datos de satisfacer necesidades explícitas e implícitas bajo determinadas condiciones de uso.

» Los clasifica estas características de calidad considerando dos puntos de vista:

- **Inherente**

- Capacidad de las características de los datos de tener el potencial intrínseco para satisfacer las necesidades explícitas e implícitas
- Este punto de vista está más relacionado con los aspectos del dominio gestionados por los expertos del negocio.

- **Dependiente del sistema:**

- Capacidad del sistema informático de alcanzar y preservar la calidad de los datos cuando los datos se utilizan en determinadas condiciones
- Este punto de vista suele ser responsabilidad de los técnicos del sistema.



MODELO DE CALIDAD DE DATOS - ISO/IEC 25012

»Inherente

- Exactitud
 - Los datos representan de forma correcta el verdadero valor
- Completitud
 - Los datos tiene valores para todos los atributos esperados
- Consistencia
 - Los datos están libre de contradicciones y están coherentes con el resto de los datos
- Credibilidad
 - Los usuarios consideran que los datos son creíbles
- Actualidad
 - Los datos tienen un tiempo adecuado



CALIDAD DE LOS DATOS ISO/IEC 25012

» Dependientes del sistema

- Disponibilidad
 - Los datos pueden ser recuperados por los usuarios autorizados
- Portabilidad
 - Los datos pueden ser instalados, reemplazados o movidos de un sistema a otro
- Recuperabilidad
 - Los datos se mantiene y preservan un nivel especificado de operaciones y de calidad, incluso en caso de fallo



MODELO DE CALIDAD DE DATOS - ISO/IEC 25012

» Inherentes y dependientes

- **Accesibilidad**
 - Se puede acceder a los datos, en especial por personas con discapacidades
- **Cumplimiento**
 - Los datos se adhieren a estándares convenciones o normas
- **Confidencialidad**
 - Los datos son accesibles e interpretados por los usuarios autorizados
- **Eficiencia**
 - Los pueden ser procesados y proporcionan el nivel de rendimiento esperado
- **Precisión**
 - Los datos son exactos
- **Trazabilidad**
 - Los datos proporcionan la información necesaria para poder auditar los accesos y las modificaciones que se les han realizado
- **Compresibilidad**
 - Los datos son leídos e interpretados por los usuarios



MODELO DE CALIDAD DE DATOS - ISO/IEC 25012

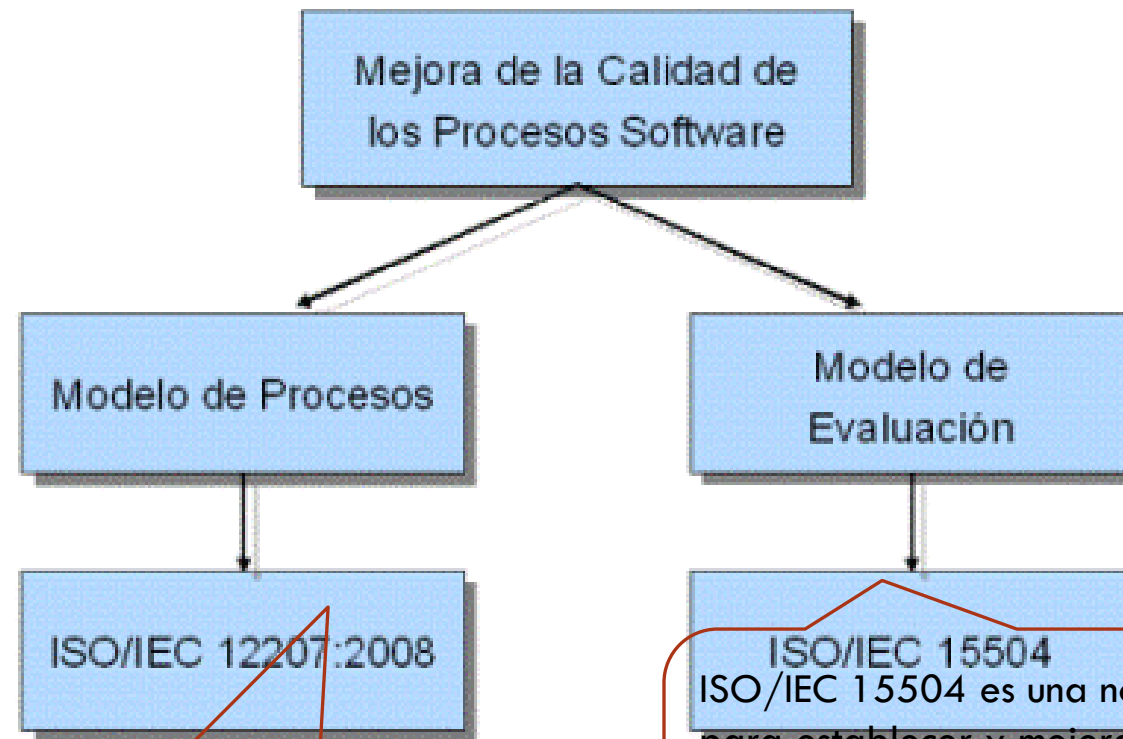
CARACTERÍSTICAS	PUNTOS DE VISTA	
	Inherente	Dependiente del Sistema
<i>Exactitud</i>	✓	
<i>Compleción</i>	✓	
<i>Consistencia</i>	✓	
<i>Credibilidad</i>	✓	
<i>Actualidad</i>	✓	
<i>Accesibilidad</i>	✓	✓
<i>Cumplimiento</i>	✓	✓
<i>Confidencialidad</i>	✓	✓
<i>Eficiencia</i>	✓	✓
<i>Precisión</i>	✓	✓
<i>Trazabilidad</i>	✓	✓
<i>Comprensibilidad</i>	✓	✓
<i>Disponibilidad</i>		✓
<i>Portabilidad</i>		✓
<i>Recuperabilidad</i>		✓

Fuente:



MODELO DE CALIDAD DE PROCESO SOFTWARE ISO/IEC 12207/15504

»Modelo de Calidad de Proceso Software



ISO/IEC 12207 establece un modelo de procesos para el ciclo de vida del software

ISO/IEC 15504
ISO/IEC 15504 es una norma internacional para establecer y mejorar la capacidad y madurez de los procesos de las organizaciones en la adquisición, desarrollo, evolución y soporte de productos y servicios

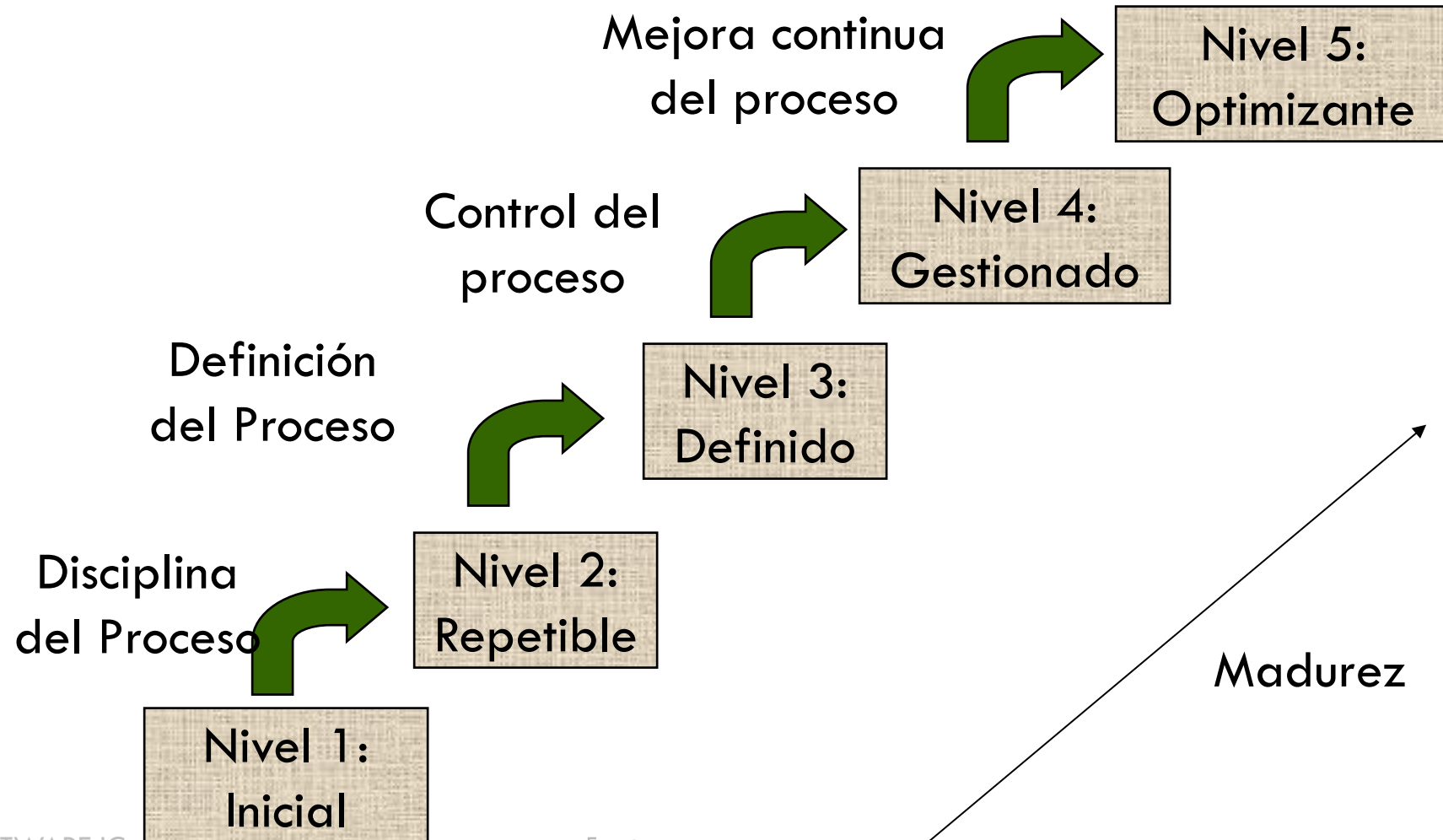
Fuente:

MODELO DE CALIDAD DE PROCESO SOFTWARE CMM - CMMI

- »En diciembre de 2000, el SEI publicó un nuevo modelo, el CMMI o "Modelo de Capacidad y Madurez - Integración", con el objetivo de realizar algunas mejoras respecto al SW-CMM (e integrarlo con el SE-CMM y el IPD-CMM, que pasaron a ser considerados como "obsoletos").
- »Incluye cuatro disciplinas, Software, Ingeniería de sistemas , Desarrollo integrado de procesos y productos y Gestión de proveedores .
- »A su vez incorpora una nueva representación, "Continua", la que permite evaluar el nivel en cada área independientemente.
- »El SEI ha desarrollado también un nuevo método de evaluación de las organizaciones según CMMI denominado SCAMPI.



MODELO DE CALIDAD DE PROCESO SOFTWARE - CMMI



ISO 9001 Y EL DESARROLLO DE SOFTWARE

» IRAM – ISO 9001:2015

- Aplicación genérica

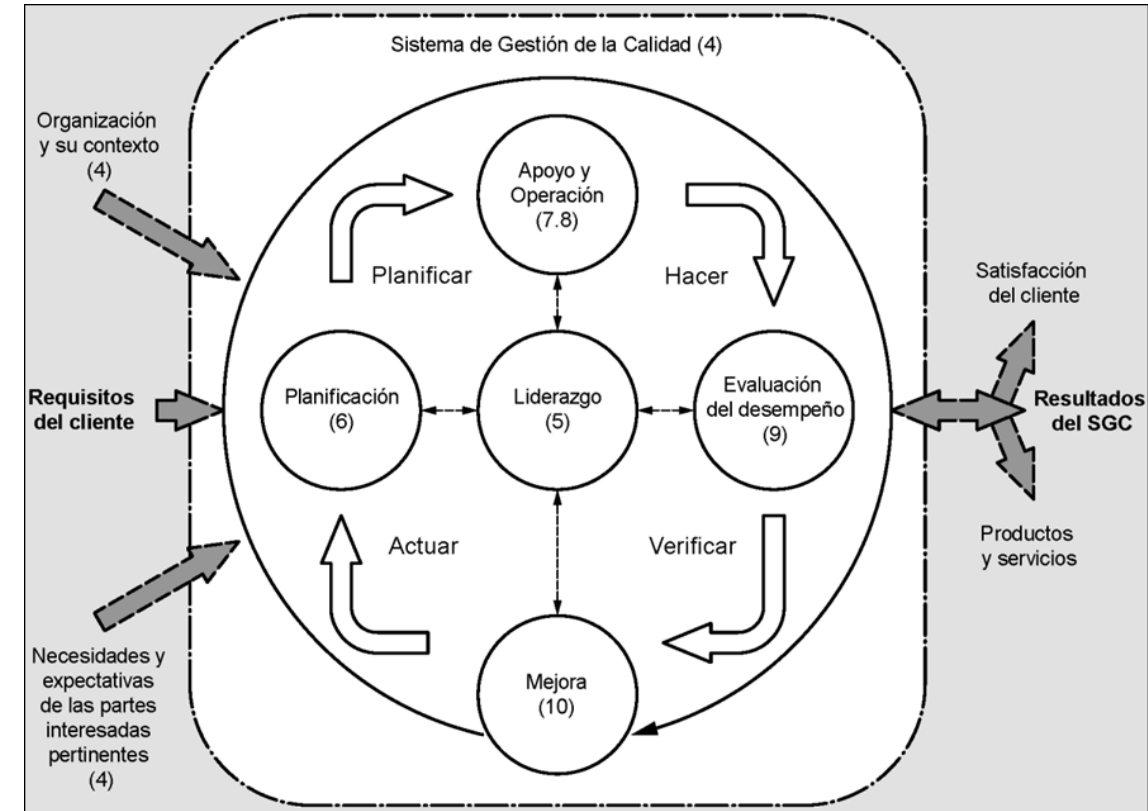
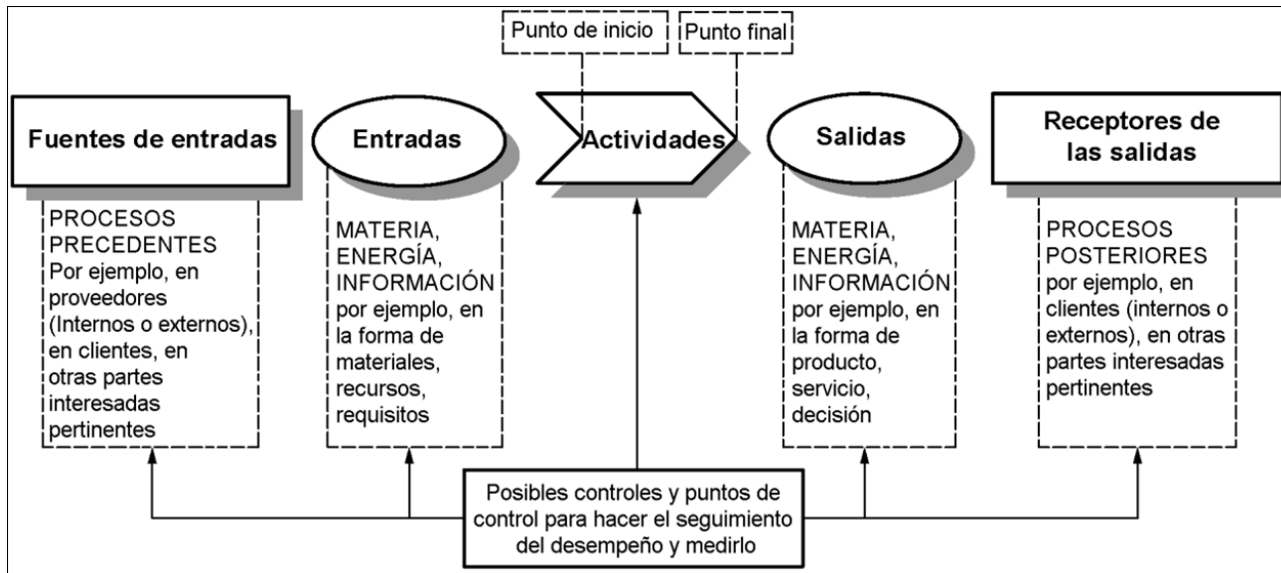
» ISO 90003:2018

» Basada ISO 9001:2015

- Directrices para la interpretación en el proceso de software
 - Proporciona una guía para identificar la evidencias dentro del proceso de software para satisfacer los requisitos de la ISO 9001

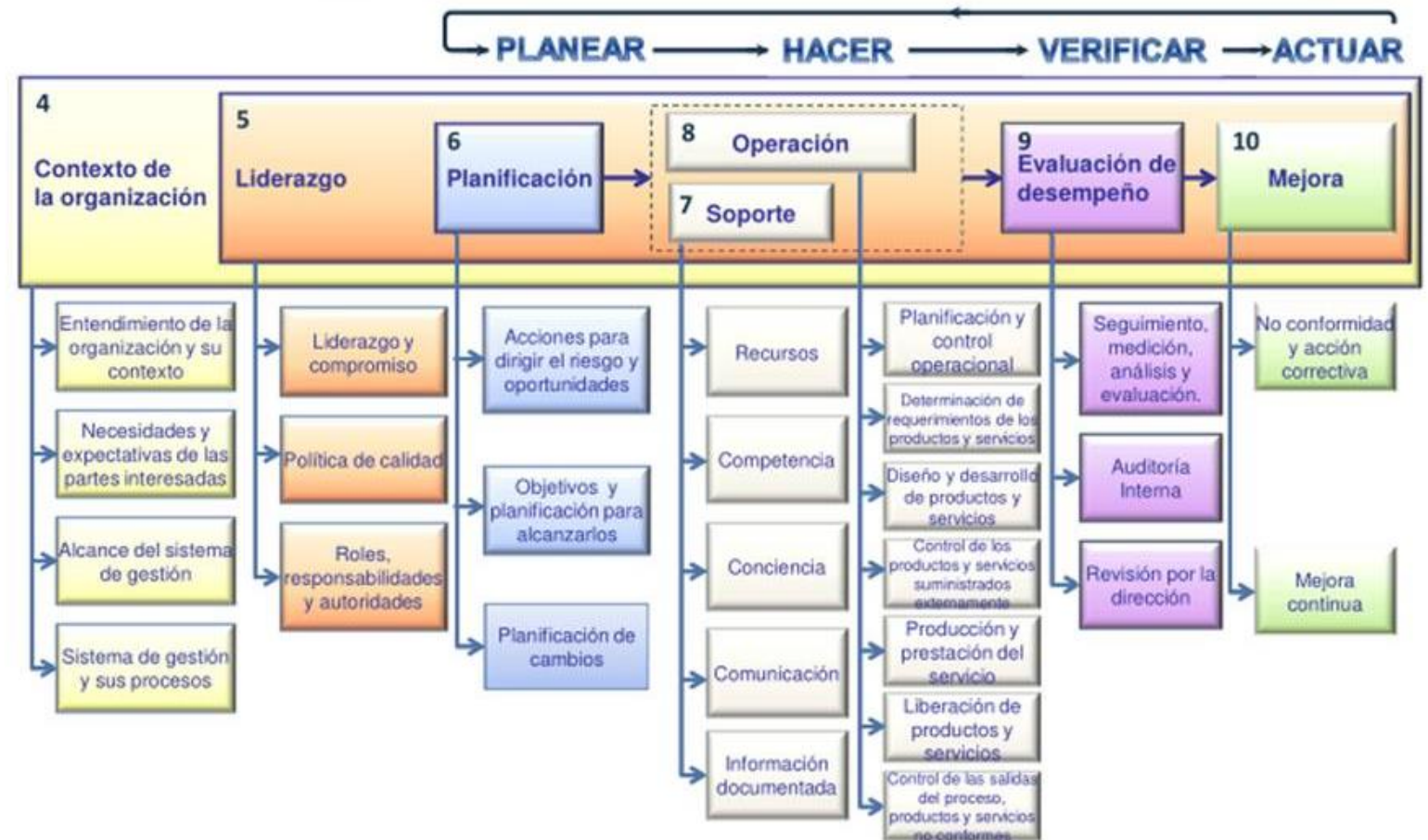


SGC – IRAM – ISO 9001:2015



SGC – IRAM – ISO 9001:2015

- 1 ALCANCE
- 2 REFERENCIAS NORMATIVAS
- 3 TÉRMINOS Y DEFINICIONES
- 4 CONTEXTO DE LA ORGANIZACIÓN
- 5 LIDERAZGO
- 6 PLANIFICACIÓN
- 7 SOPORTE
- 8 OPERACIONES
- 9 EVALUACIÓN DEL RENDIMIENTO
- 10 MEJORA



CLASIFICACIÓN DE NORMAS Y MODELOS DE CALIDAD

OTRAS NORMAS

CALIDAD DE PRODUCTO DE SOFTWARE
CALIDAD DE USO – CALIDAD DE DATOS
ISO/IEC 9126 /14598 - ISO/IEC 25000

CALIDAD DE SERVICIOS
ISO/IEC 20000 - ITIL



Orientadas a PyMEs

PMBOOK - SWEBOOK- SIX SIGMA - ISO/IEC 12207 - ISO/IEC 15504 –
ISO/IEC 90003 -CMMI – SCAMPI – IDEAL - ISO/IEC 29110



CALIDAD TOTAL – TQM – ISO/IEC 9001
SEGURIDAD DE LA INFORMACIÓN – ISO/IEC 27001

RESUMEN

Gestión de la Configuración del Software (GCS)

- Elementos de la configuración
- Línea Base
- Proceso de la GCS

Métricas

- Definiciones
- Producto - Líneas de Código
- Control o predicción - Punto función
- GQM

Estimaciones

- Definiciones
- Juicio experto
- Técnica Delphi
- División de trabajo
- COCOMO I/II

Calidad de Software

- Definiciones
- Modelos holístico de la calidad
- Calidad de Producto
- Calidad de Procesos
- Estándares
 - ISO/IEC 25000/ 12207/15504/29110/ 9001/90003
 - CMMI