INGENIERÍA DE SOFTWARE

MÉTRICAS
CALIDAD DE SOFTWARE

EN CLASES ANTERIORES VIMOS . . .

Conceptos generales

Modelos proceso

Metodologías agiles

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

Gestión de la Configuración del Software (GCS)

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

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.

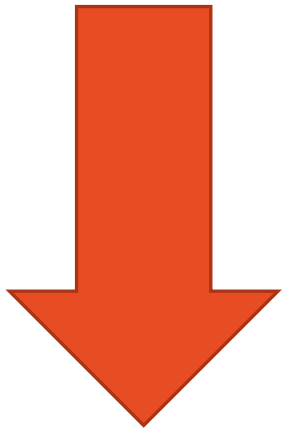
»Indicador:

- 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

Fuente: Pressman Cap. 15

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

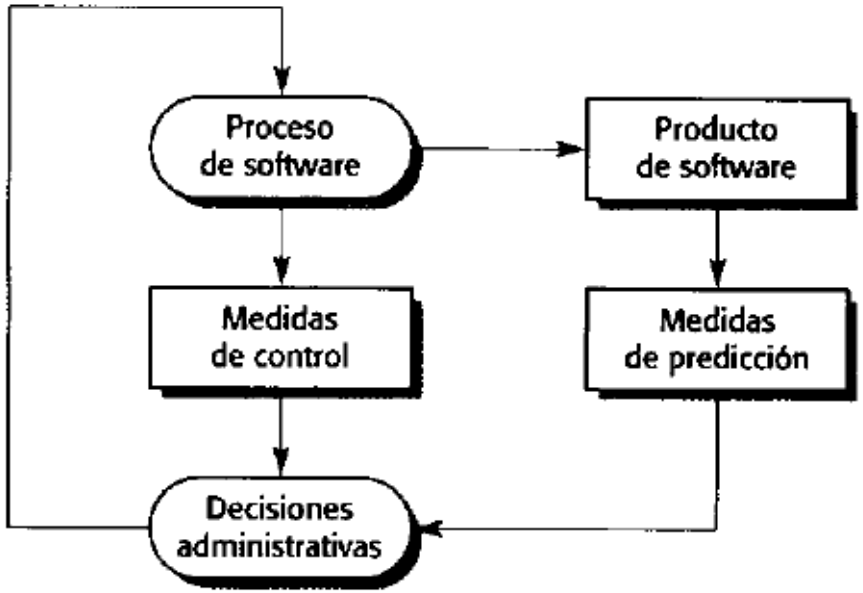
»Métricas de predicción (del producto)

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

Fuente: Clasificación de las métricas

CLASIFICACIÓN DE LAS MÉTRICAS

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



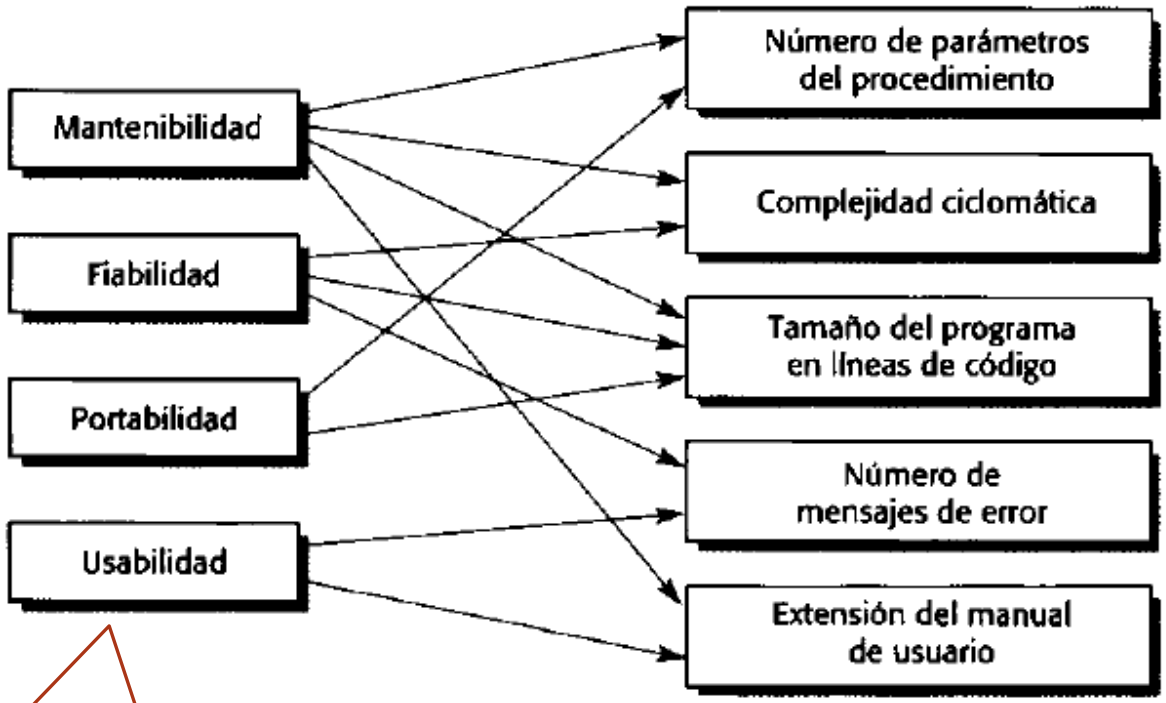
Fuente:

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

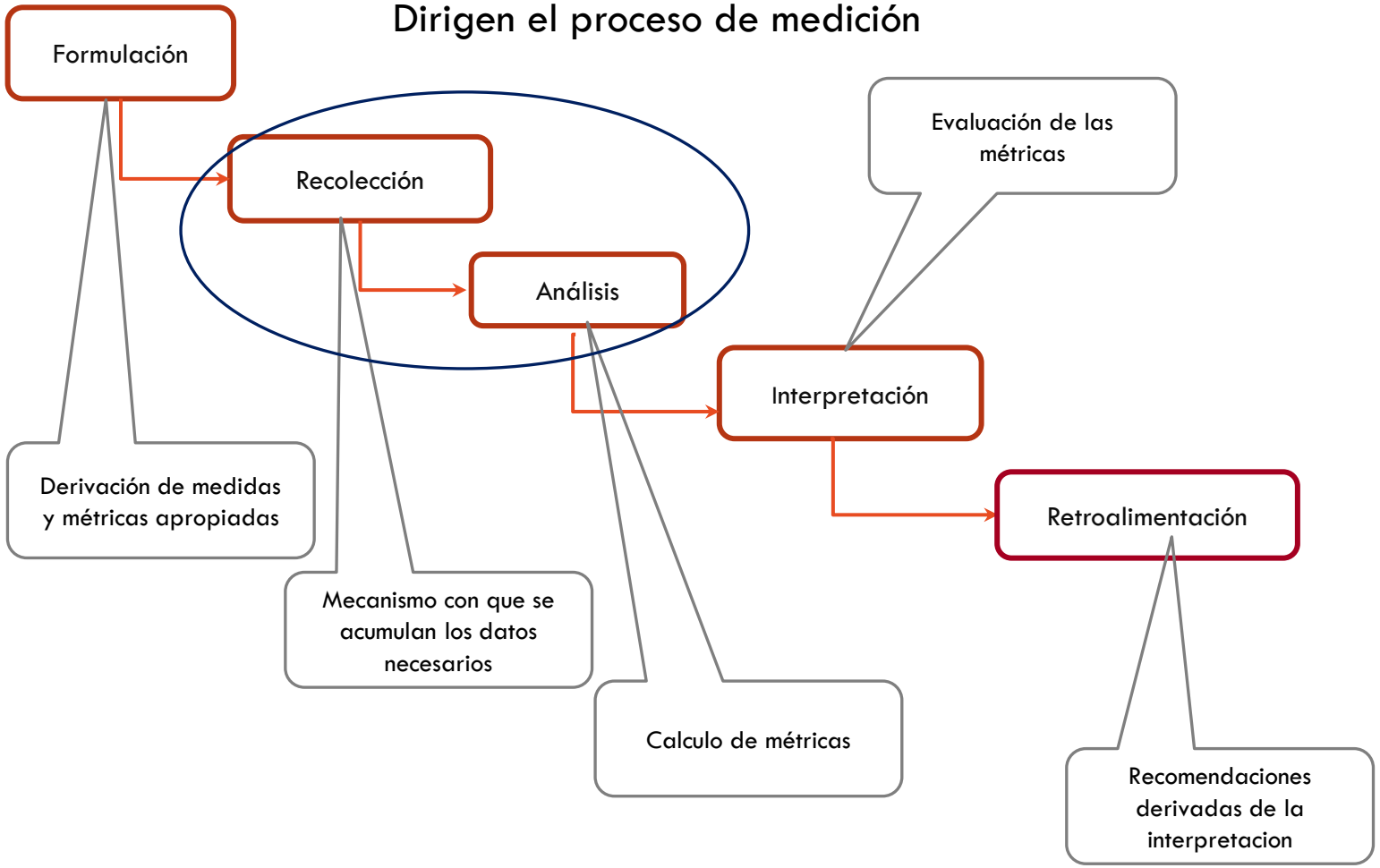
La relación entre atributos internos y externos debe comprenderse, validarse y expresarse en términos de una formula o modelo.

Fuente:

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



Fuente:

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

Fuente:

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 = KLDC/persona-mes
Calidad = errores/KLDC
Documentación = págs.. Doc./ KLDC
Costo = $/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 = Cantidad de líneas de comentarios
- Luego:
  - 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$S	Errores	Personas-mes	Errores/KLDC	U$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 = KLDC/persona-mes
Calidad = errores/KLDC
Documentación = págs.. Doc./ KLDC
Costo = $/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)?

Fuente:

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 descripto 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(Fi)] i=1 a 14 0 <= Fi<=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

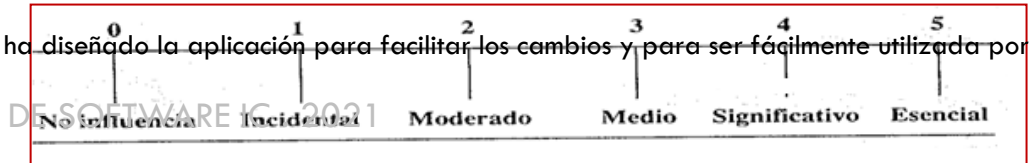
Son valores de ajuste de la complejidad según las preguntas de la siguiente pantalla

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

Fuente:

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

Productividad = PF / Persona_mes
Calidad = Errores / PF
Costo = $ / PF

Fuente: Pressman Cap. 15

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)

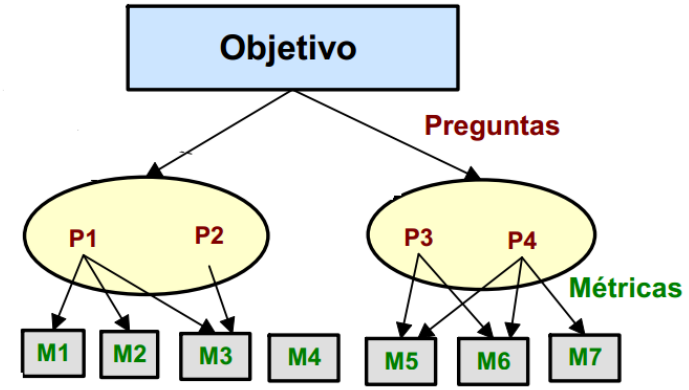
- Dicho método esta orientado a lograr una métrica que "mida" cierto objetivo . El mismo nos permite mejorar la calidad de nuestro proyecto.

- 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

Fuente:

GQM EJEMPLO

»Indicadores

Nombre	**Descripción**	**Fórmula**
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

Fuente:

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.

Fuente: Fijación de precio

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

Fuente:

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)

» (COnstructive COst MOdel, modelo constructivo de costos) se obtuvo recopilando datos de varios proyectos grandes.

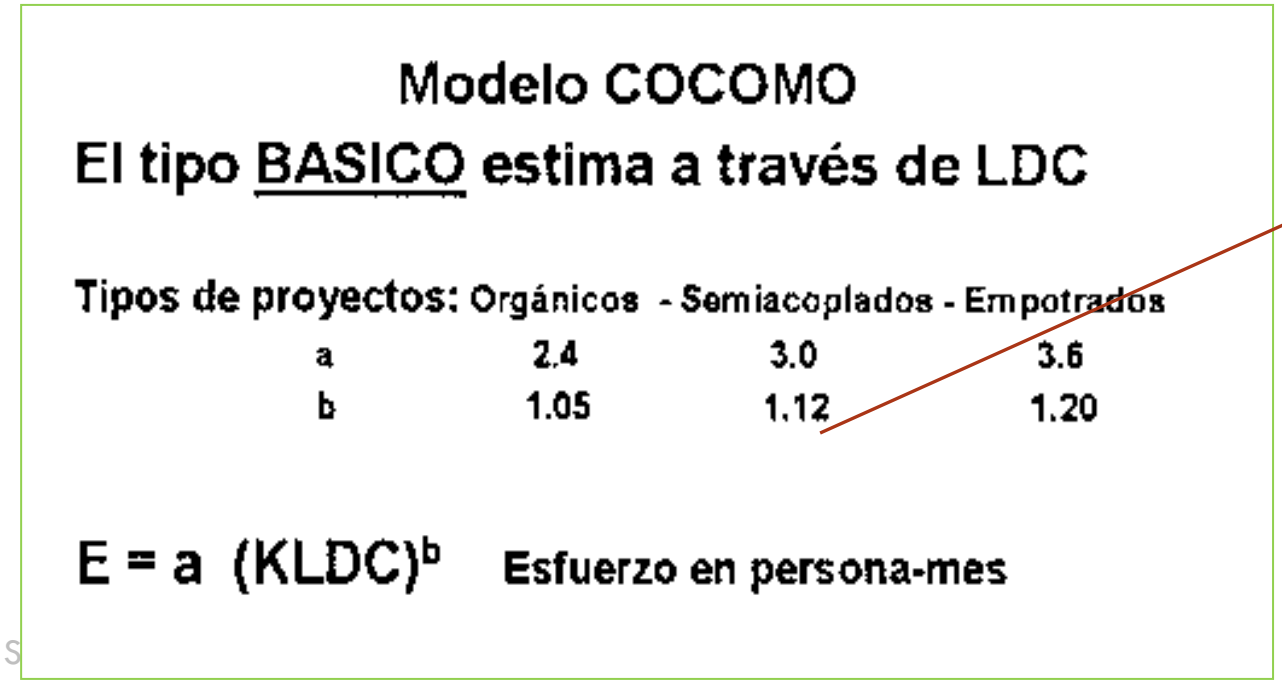
»Las formulas que utiliza el COCOMO vinculan el tamaño del sistema y del producto, factores del proyecto y del equipo con el esfuerzo necesario para desarrollar el sistema.

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



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)

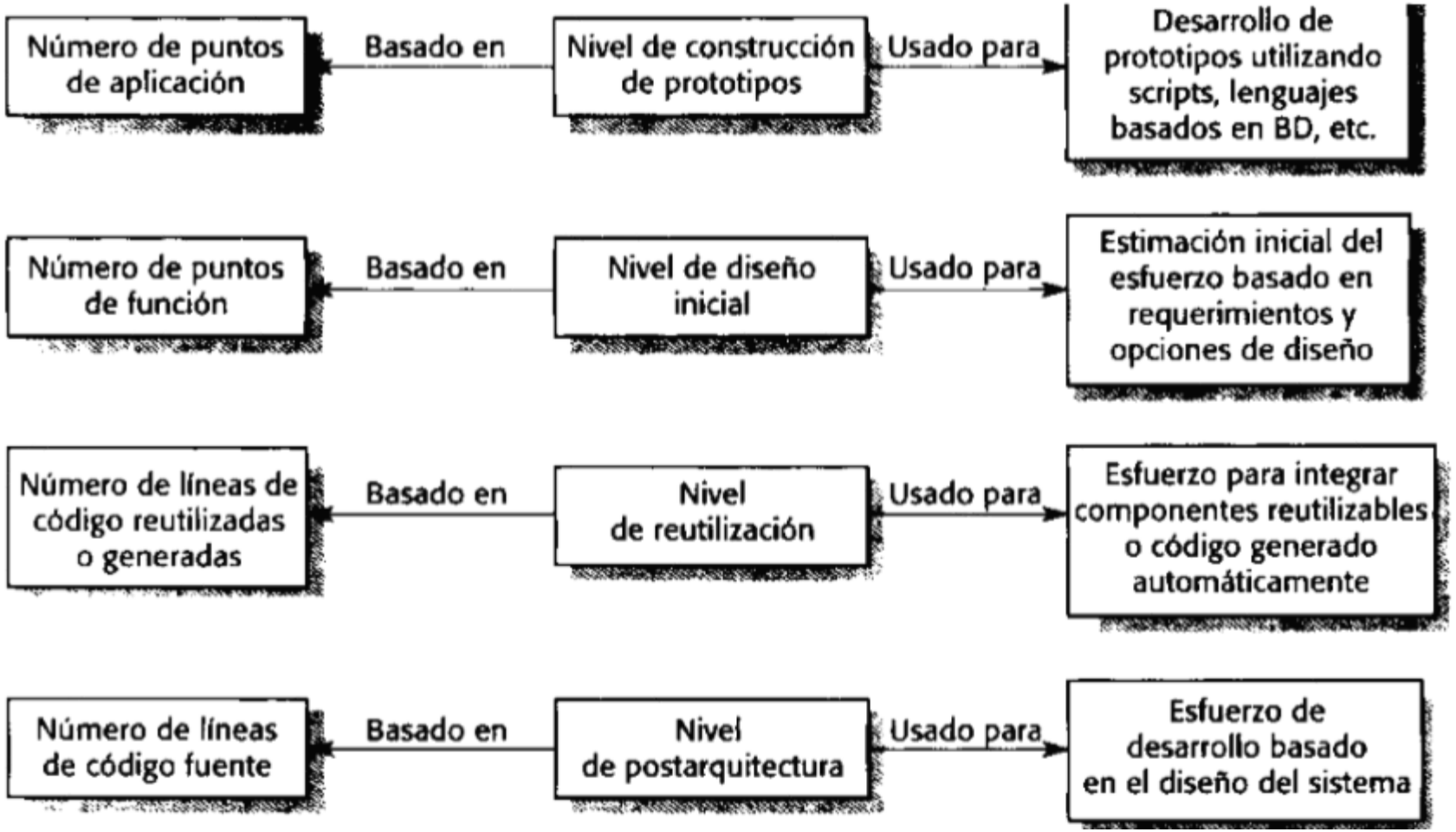
» Reconoce que las líneas de código son difíciles de estimar tempranamente. Considera diferentes enfoques para el desarrollo , como la construcción de prototipos, el desarrollo basado en componentes, desarrollo en espiral y engloba varios niveles que producen estimaciones detalladas de forma incremental.

» COCOMO II está compuesto por 4 niveles:

- De construcción de prototipos
- De diseño inicial
- De reutilización
- De post-arquitectura

Fuente: Fuente: Somerville Cap. 26

NIVELES DE COCOMO II



Fuente:

COCOMO II 1. NIVEL CONSTRUCCIÓN DE PROTOTIP

Punto de aplicación = Punto objeto

»La fórmula para el cálculo del esfuerzo para el pr ado del sistema es:

»PM = (NAP x (1-%reutilización/100))/PROD

»PM = esfuerzo estimado en personas/mes

»NAP = total de puntos de a

»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

Fuente: Fuente: Somerville Cap. 26

COCOMO II 2. NIVEL DE DISEÑO INICIAL

- La fórmula para las estimaciones en este nivel es:

- Esfuerzo = A x TamañoB x M.

- Boehm propone que A = 2.94. (Otros autores proponen: 2.45)
- Tamaño = KLDC (miles de líneas de código fuente).
- B = 0.91 X Σ SFj (ver tabla)

- M = PERS x RCPX x RUSE x PDIF x PREX x FCIL x SCED (ver mas adelante)

Fuente: Fuente: Somerville Cap. 26

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)

Fuente: Fuente: Somerville Cap. 26

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).

Fuente: Fuente: Somerville Cap. 26

COCOMO II - 3. NIVEL DE REUTILIZACIÓN

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

- PMauto = (ASLOC x 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

Fuente:

COCOMO II - 4. NIVEL DE POST-ARQUITECTURA

- Las estimaciones están basadas en la misma fórmula básica
- PM = A x TamañoB x 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

Fuente: Fuente: Somerville Cap. 26

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

1. DEFINICIÓN DE CALIDAD

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.

»calidad

- (Del la

1.f. Pro "Esta tela es de b

2.f. Bue do los mercad

3.f. Carácter, genio, índole.

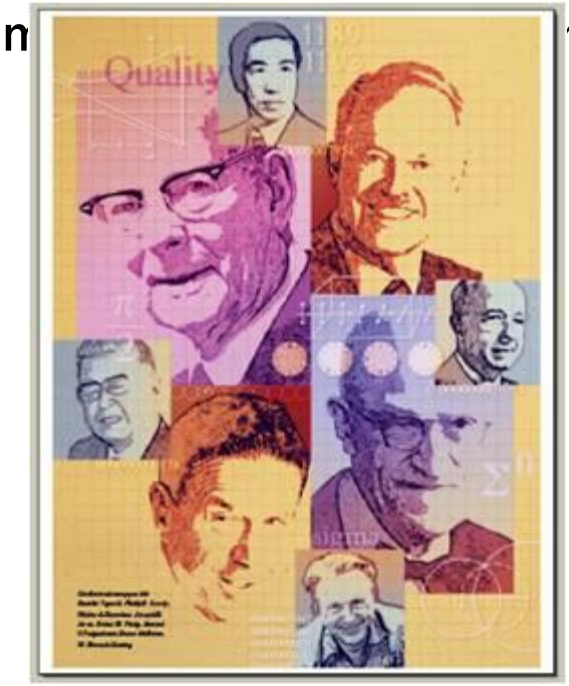
4.f. Condición o requisito que se pone en un contrato.

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

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 llam rús de la Calidad.



Fuente: 2. Que es 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.

Fuente: ¿Que es la Calidad?

¿QUE ES LA CALIDAD?

»Reconocimiento del mercado - Autos

Fuente:

¿QUE ES LA CALIDAD?

»Reconocimiento del mercado - software

¿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 implicitas" (ISO 8402)

¿QUE ES LA CALIDAD?

¿Qué tiene más calidad?

CALIDAD DE LOS SISTEMAS DE INFORMACIÓN

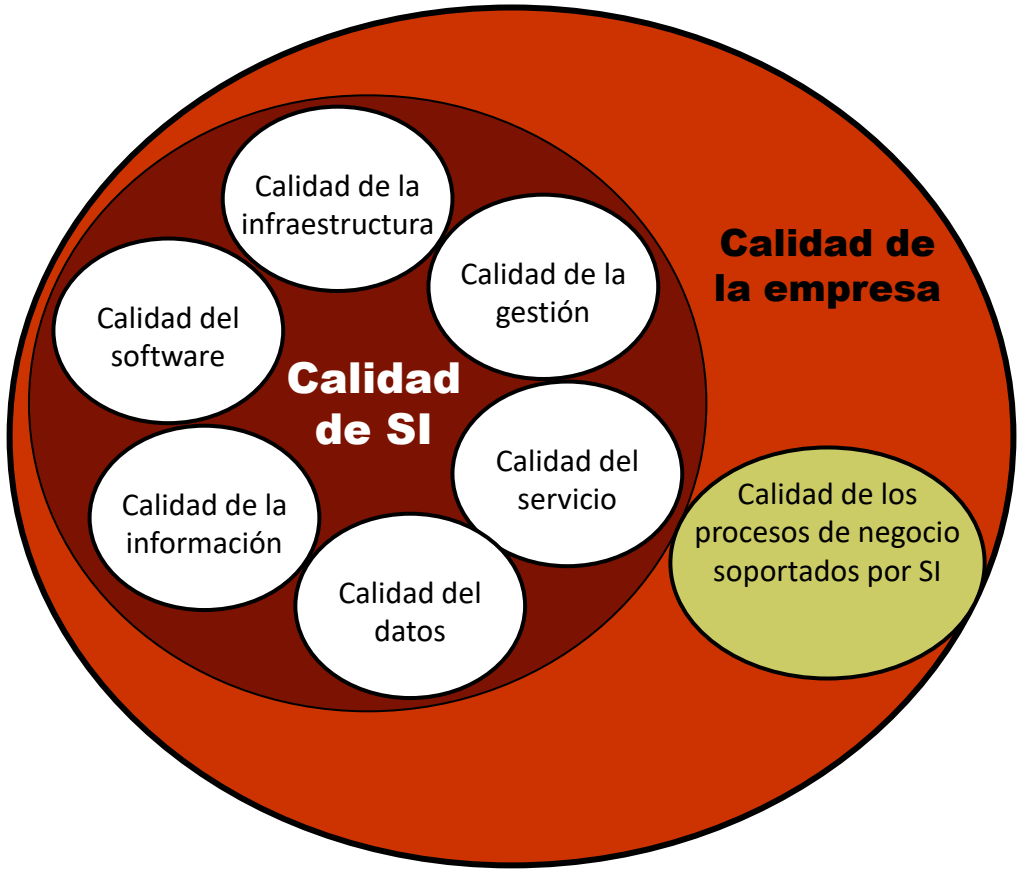
- La importancia de los sistemas de información (SI) en la actualidad hace necesario que las empresas de tecnología hagan mucho hincapié en los estándares de calidad.
- Stylianou y Kumar plantean que se debe apreciar la calidad desde un todo, donde cada parte que la componen debe tener su análisis de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

Fuente: Calidad de los Sistemas de Información

COMPONENTES

Visión holística de la calidad

Stylianou y Kumar (2000)



Fuente:

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

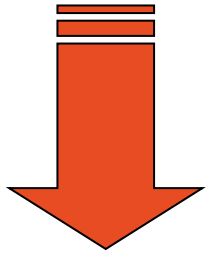
Fuente:

CALIDAD DE SOFTWARE

»La calidad del software se ha mejorado significativamente en esto 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.

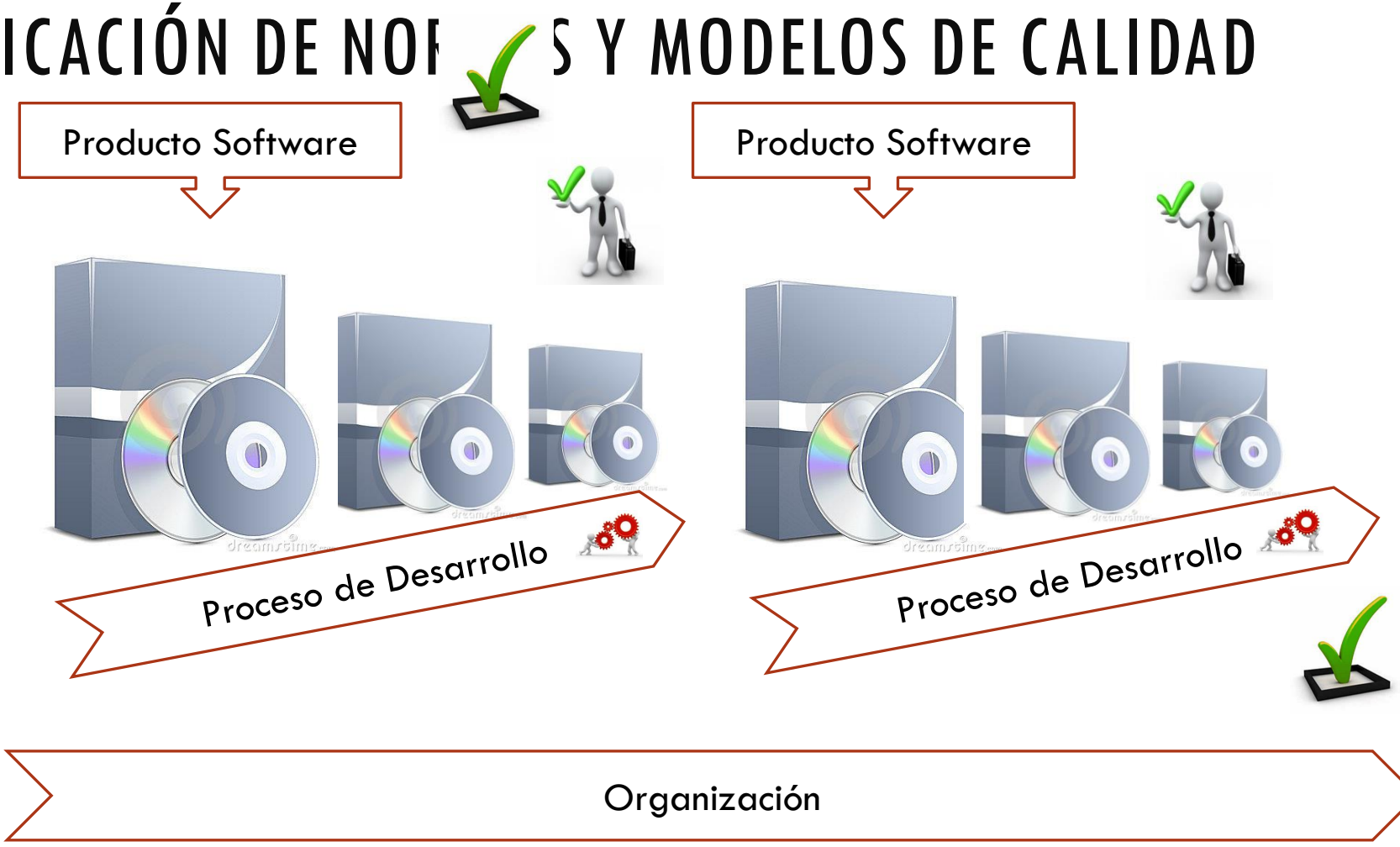
»Proceso

- 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.

Fuente: Calidad del Producto y Proceso

CLASIFICACIÓN DE NOR S Y MODELOS DE CALIDAD



Fuente:

CLASIFICACIÓN DE NORMAS Y MODELOS DE CALIDAD

Servicio

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

Fuente:

CLASIF RMAS Y MODELOS DE CALIDAD

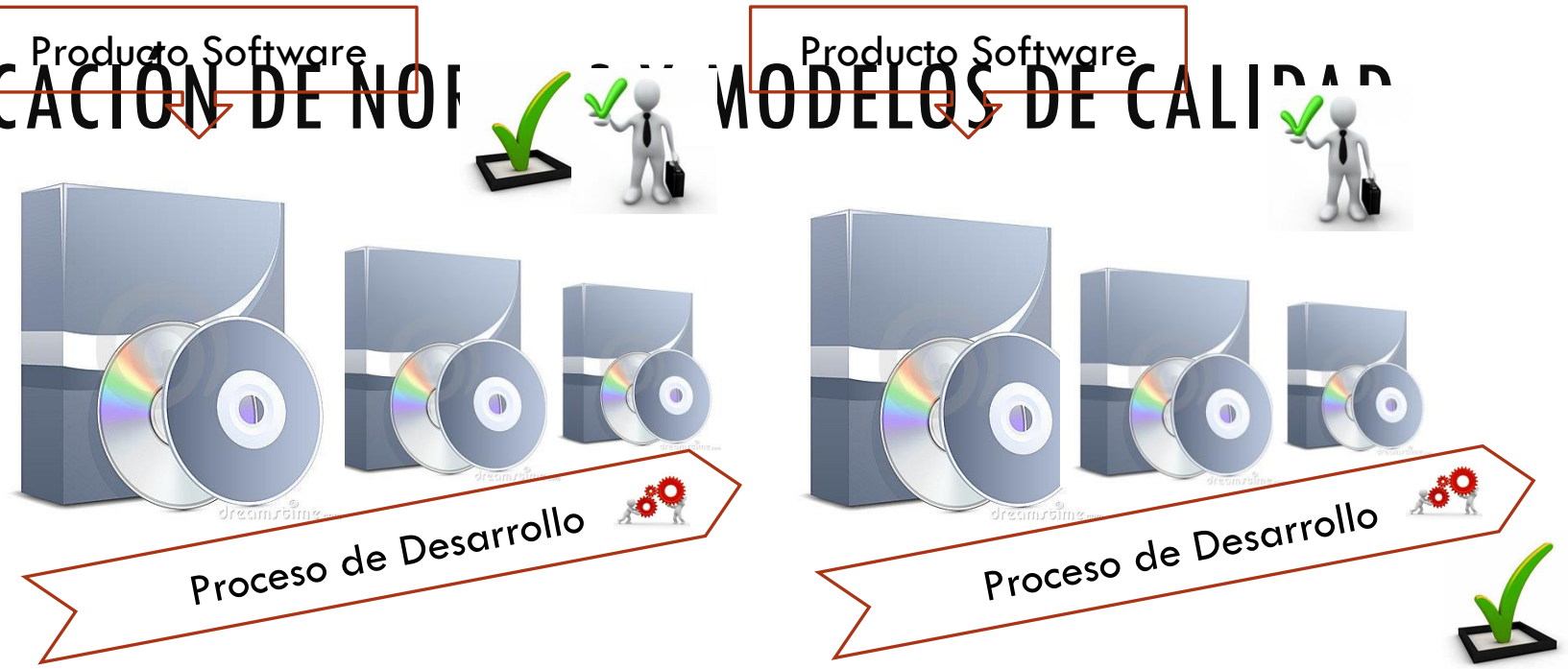
Proceso de Desarrollo

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

MPS-BR - MOPROSOFT -COMPETISOFT
METRICA V3 - ISO/IEC 29110

Fuente:

CLASIFICACIÓN DE NOR[illegible] MODELOS DE CALIDAD



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 90003 -CMMI – SCAMPI – IDEAL -MPS-BR - MOPROSOFT - COMPETISOFT METRICA V3 - ISO/IEC 29110

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

Fuente:

MODELO DE CALIDAD SQUARE ISO/IEC 25000

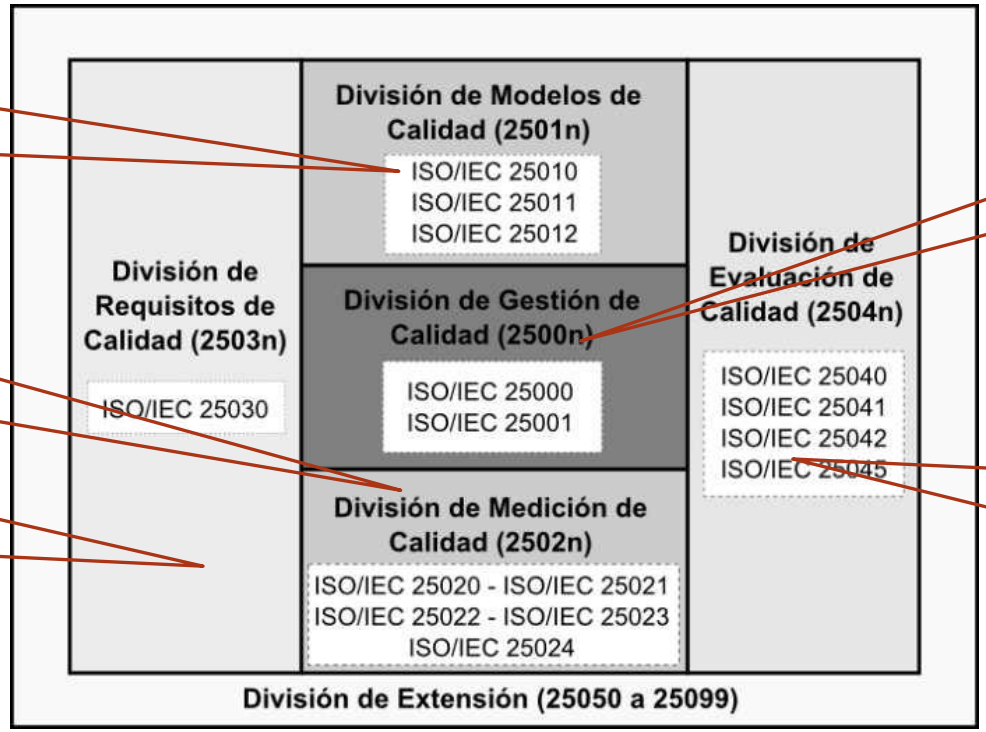
»ISO/IEC 25000 SQuaRE Software product Quality Requeriment and Evaluation

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 practicas 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.

Fuente: SQuaRE cubre tres procesos de calidad complementarios:

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
  - 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

Fuente:

MODELO DE CALIDAD SQUARE ISO/IEC 25010

Fuente:

MODELO DE CALIDAD SQUARE ISO/IEC 25000

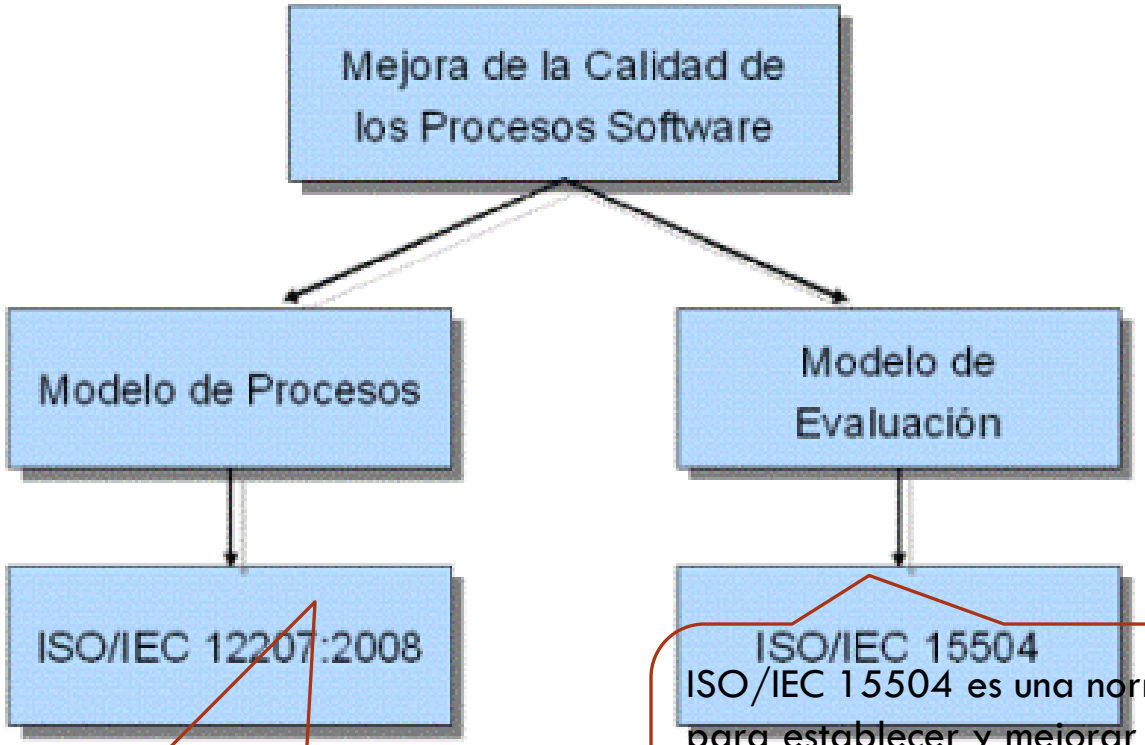
»Ejemplo de métricas de Interoperabilidad

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 measure-ment	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<=X<= 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 *It is recommended to test specified data transaction.*									
Data exchangeability (User's success attempt based)	How often does the end user fail to exchange data between target software and other software? How often are the data transfers between target software and other software successful? Can user usually succeed in exchanging data?	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 b) Y= A / T T= Period of operation time	0<=X<= 1 The closer to 1.0 is the better. 0<=Y The closer to 0, is the better.	a) Absolute b) Ratio	A= Count B= Count X= Count/ Count Y= Count/ Time T= Time	Req. spec. (User manual) Test report	5.4 Operation	Maintainer

MODELO DE CALIDAD SOFTWARE

»Modelo de Calidad de Proceso Software



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

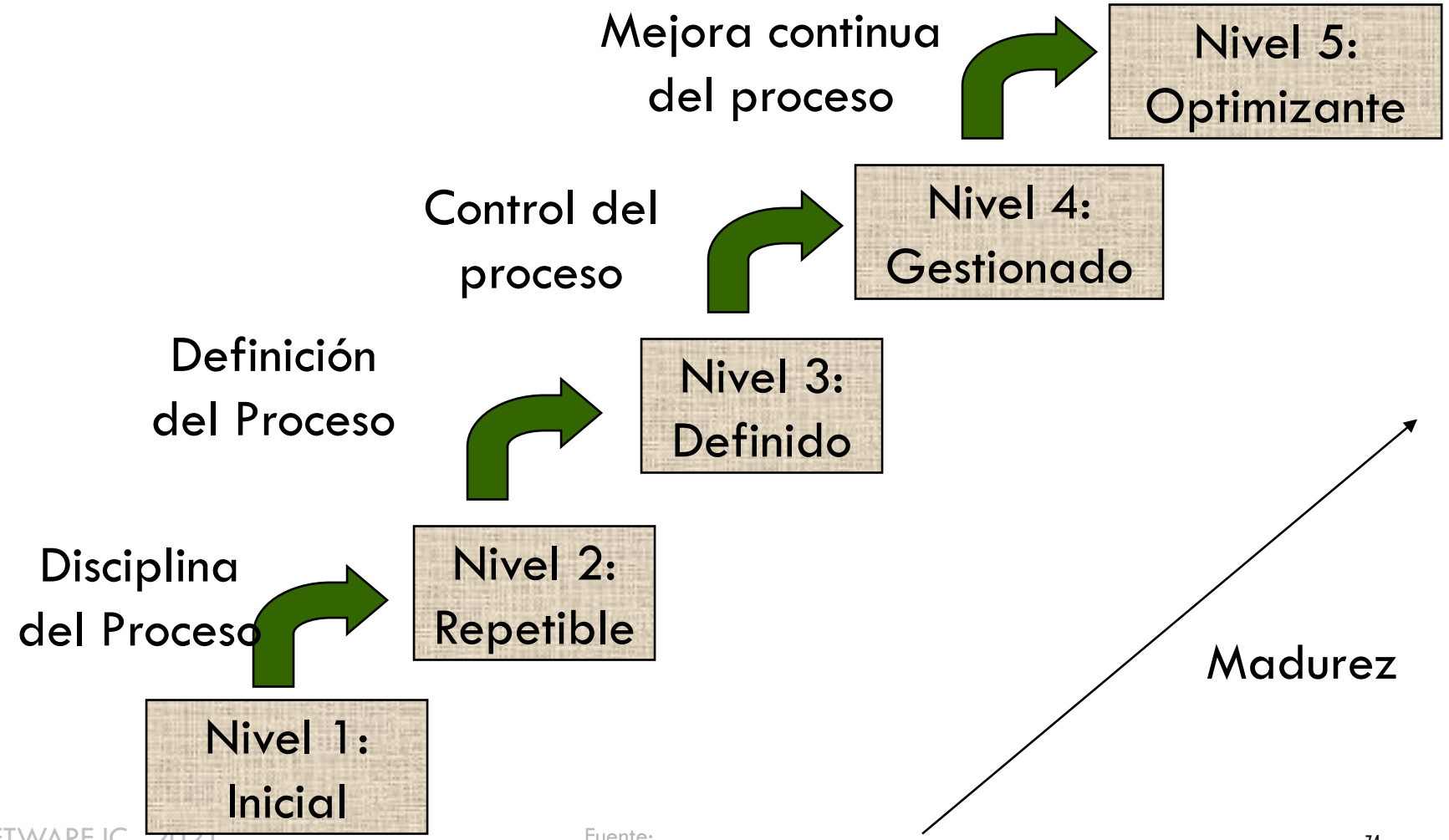
ISO/IEC 15504 es una norma internacional para establecer y mejorar la capacidad y madurez de los procesos de las organizaciones en la adquisición, desarrollo, evolución y soporte de productos y servicios

Fuente:

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.

CMMI


Mejora continua del proceso
Nivel 5: Optimizante
Control del proceso
Nivel 4: Gestionado
Definición del Proceso
Nivel 3: Definido
Disciplina del Proceso
Nivel 2: Repetible
Nivel 1: Inicial
Madurez

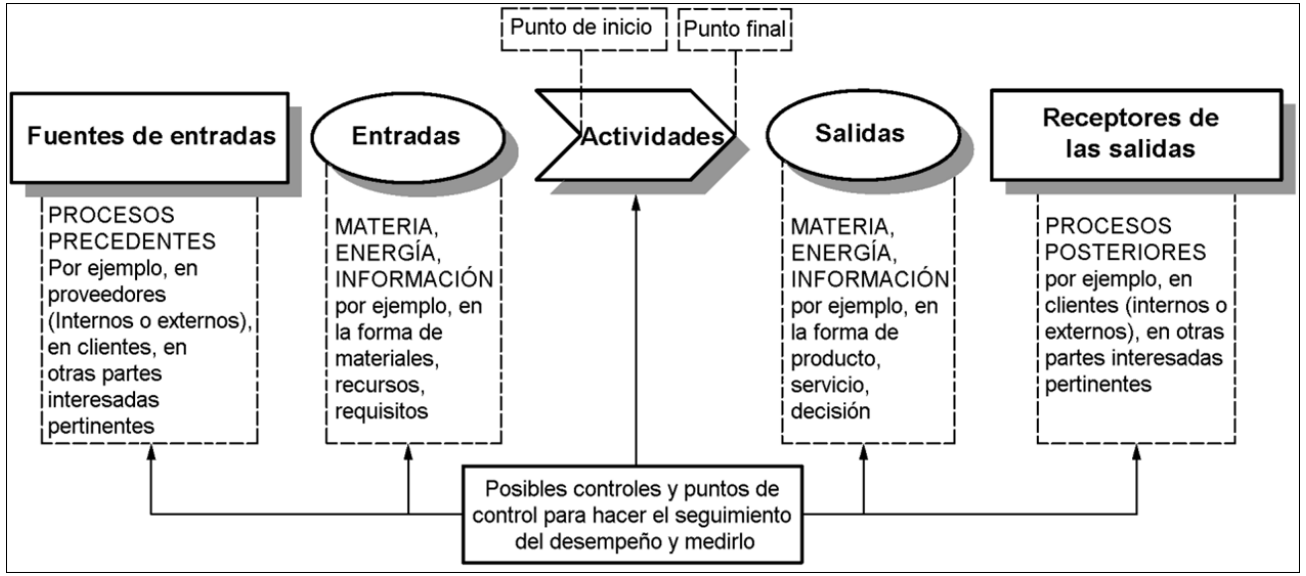
Fuente:

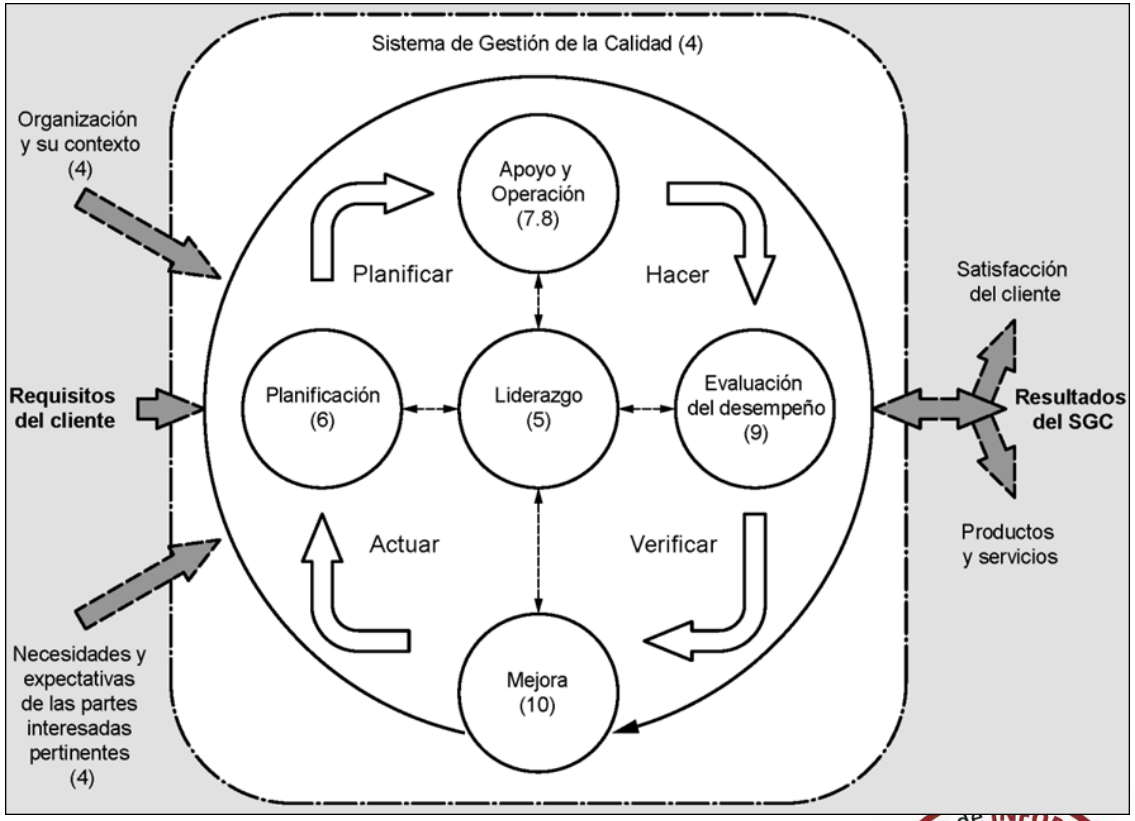
ISO 9001 Y EL DESARROLLO DE SOFTWARE

- IRAM – ISO 9001:2015
  - Aplicación genérica
- ISO 90003:2004 (nueva versión en desarrollo)
  - Basada ISO 9001:2000 (se espera una actualización para el próximo año)
  - 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

Fuente: ISO 9000 y El desarrollo de software

SGC – IRAM – ISO 9001:2015

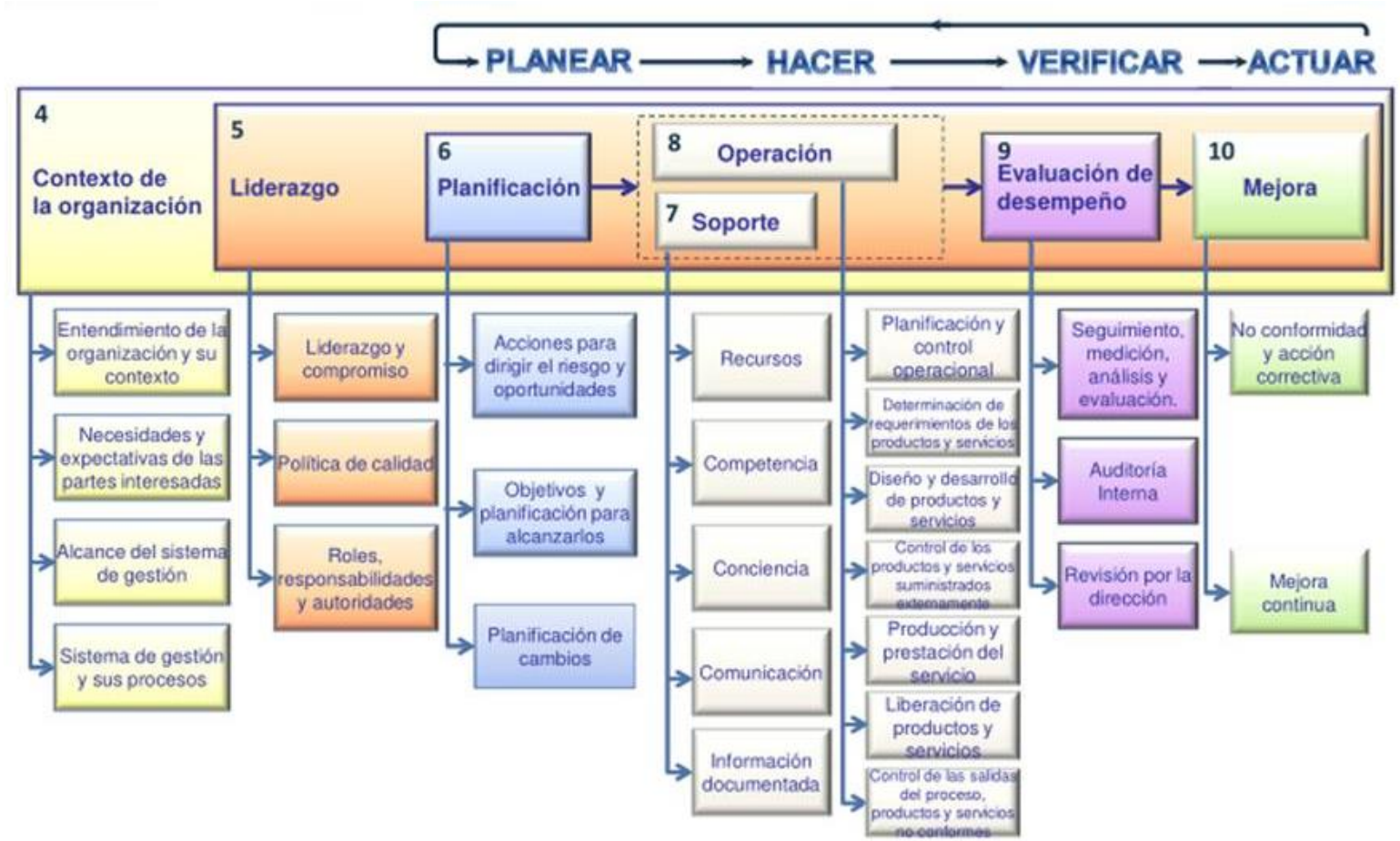




Fuente:

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



Fuente:

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 -**MPS-BR - MOPROSOFT - COMPETISOFT METRICA V3 - ISO/IEC 29110**

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

Fuente:

RESUMEN

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