Ingeniería de software II

Gestión de Proyectos

Gestión de Proyectos

Métricas

Elementos clave de la gestión de proyectos

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

} Métricas y Estimaciones

Métricas

Clave tecnológica

Objetivos fundamentales:

- Entender
- Controlar
- Mejorar
- Evaluar

Métricas – Definiciones

Medida 	Medición 
Métrica 	Indicador 

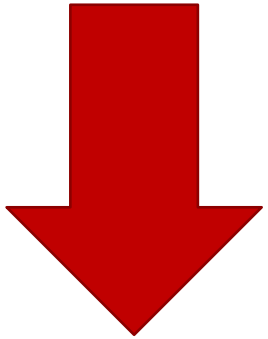
LOC por punto de función

Lenguaje	LOC/FP	Lenguaje	LOC/FP
Ensamblador	320	Basic ANSI/Quick/Turbo	64
Macroensamblador	213	Java	53
C	150	Visual C++	34
Fortran	106	Fogeo 2,5	34
Cobol	106	Visual Basic	32
Pascal	91	Delphi	29
Cobol ANSI 85	91	C++	29
Basic	91	Visual Cobol	20
RPG	80	Clipper	19
PLR	80	Power Builder	16
Ada	71	Hoja de Calculo	6

»Pressman Cap. 23

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

Procesos/Proyectos Software/Productos

Métricas del proyecto

Propósitos tácticos

Uso

- Ajustes en el calendario y evitar demoras
- Valorar el estado de un proyecto en marcha
- Rastrear riesgos
- Descubrir áreas de problemas
- Ajustar flujo de trabajo/tareas
- Evaluar habilidad del equipo.

Métricas del proceso

- Propósitos estratégicos

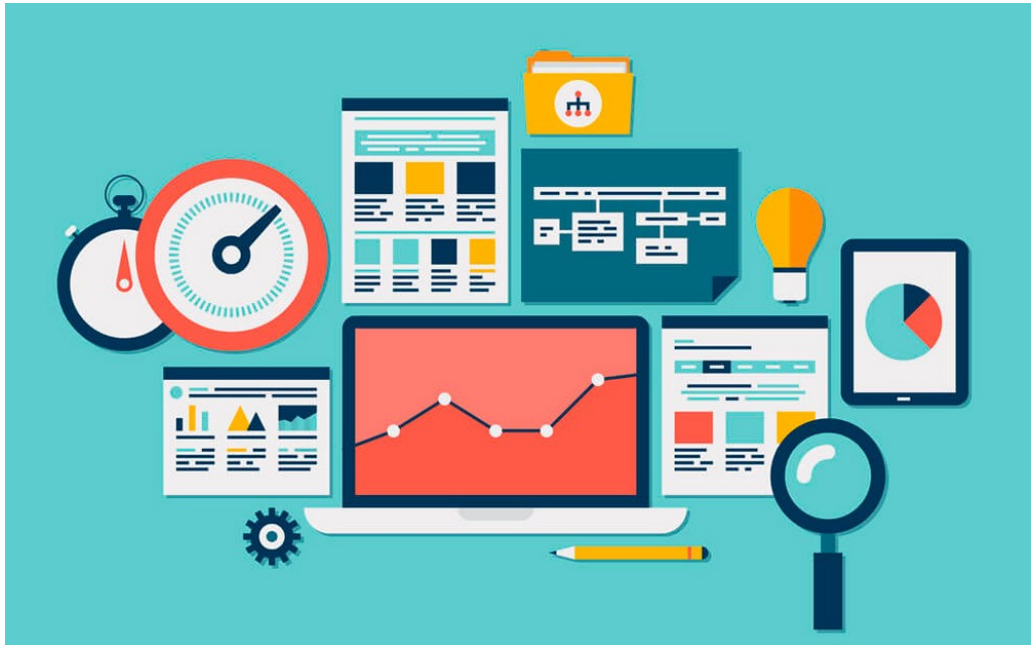
Recopilación a través de todos los proyectos y por un espacio de tiempo.

Intención proporcionar un conjunto de indicadores para mejorar el proceso.

Métricas del proceso

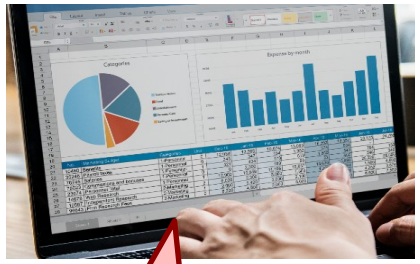
Uso

- Sentido común y sensibilidad organizacional.
- Retroalimentación.
- No usar métricas para valorar a los individuos.
- Establecer metas y métricas claras.
- No excluir métricas.



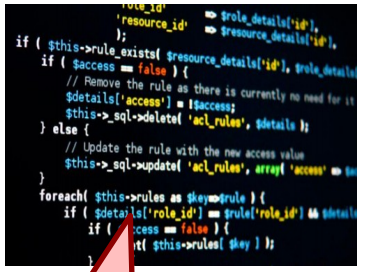
Métricas del producto

Cómo los medimos ?



Dinámica
- Hec un programa en ejecución.
- Ay valorar la eficiencia y

Se relacionan con las características de calidad del software



Estáticas
- Hec epresentaciones del

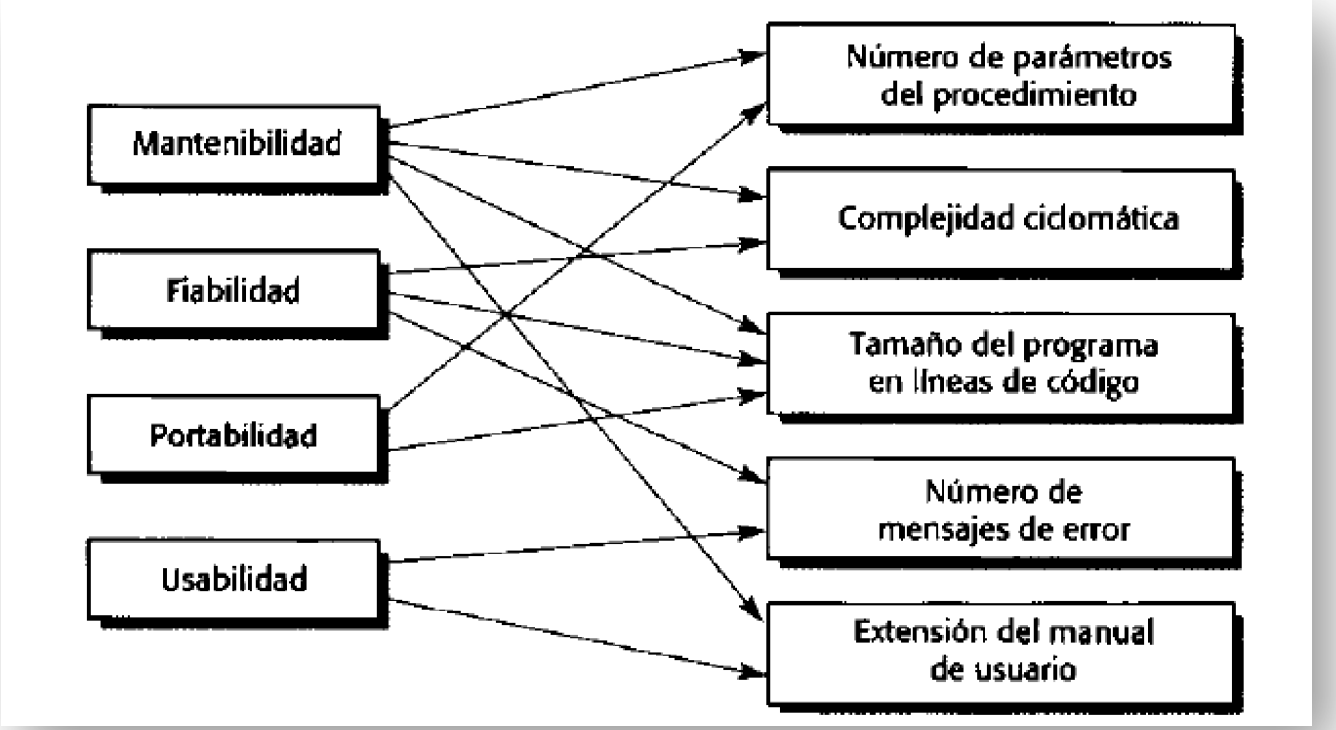
d.

Se relacionan de manera indirecta con los atributos de calidad del software

Fuente:

Métricas del producto

Atributos de calidad externos vs Atributos internos



Métricas estáticas del producto

Fan-in/Fan-out	Fan-in es una medida del número de funciones o métodos que llaman a otra función o método (por ejemplo, X). Fan-out es el número de funciones que son llamadas por una función X. Un valor alto de fan significa que X está fuertemente acoplada al resto del diseño y que los cambios en X tendrán muchos efectos importantes. Un valor alto de fan-out sugiere que la complejidad de X podría ser alta debido a la complejidad de la lógica de control necesaria para coordinar los componentes llamados.
Longitud del código	Ésta es una medida del tamaño del programa. Generalmente, cuanto más grande sea el tamaño del código de un componente, más complejo y susceptible de errores será el componente. La longitud del código ha mostrado ser la métrica más fiable para predecir errores en los componentes.
Complejidad ciclomática	Ésta es una medida de la complejidad del control de un programa. Esta complejidad del control está relacionada con la comprensión del programa.
Longitud de los identificadores	Es una medida de la longitud promedio de los diferentes identificadores en un programa. Cuanto más grande sea la longitud de los identificadores, más probable será que tengan significado; por lo tanto, el programa será más comprensible.
Profundidad del anidamiento de las condicionales	Ésta es una medida de la profundidad de anidamiento de las instrucciones condicionales «if» en un programa. Muchas condiciones anidadas son difíciles de comprender y son potencialmente susceptibles de errores.
Índice de Fog	Ésta es una medida de la longitud promedio de las palabras y las frases en los documentos. Cuanto más grande sea el índice de Fog, el documento será más difícil de comprender.

Métricas estáticas del producto

Métricas OO

Métrica orientada a objetos	Descripción
Métodos ponderados por clase (*weighted methods per class*, WMC)	Éste es el número de métodos en cada clase, ponderado por la complejidad de cada método. Por lo tanto, un método simple puede tener una complejidad de 1, y un método grande y complejo tendrá un valor mucho mayor. Cuanto más grande sea el valor para esta métrica, más compleja será la clase de objeto. Es más probable que los objetos complejos sean más difíciles de entender. Tal vez no sean lógicamente cohesivos, por lo que no pueden reutilizarse de manera efectiva como superclases en un árbol de herencia.
Profundidad de árbol de herencia (*depth of inheritance tree*, DIT)	Esto representa el número de niveles discretos en el árbol de herencia en que las subclases heredan atributos y operaciones (métodos) de las superclases. Cuanto más profundo sea el árbol de herencia, más complejo será el diseño. Es posible que tengan que comprenderse muchas clases de objetos para entender las clases de objetos en las hojas del árbol.
Número de hijos (*number of children*, NOC)	Ésta es una medida del número de subclases inmediatas en una clase. Mide la amplitud de una jerarquía de clase, mientras que DIT mide su profundidad. Un valor alto de NOC puede indicar mayor reutilización. Podría significar que debe realizarse más esfuerzo para validar las clases base, debido al número de subclases que dependen de ellas.
Acoplamiento entre clases de objetos (*coupling between object classes*, CBO)	Las clases están acopladas cuando los métodos en una clase usan los métodos o variables de instancia definidos en una clase diferente. CBO es una medida de cuánto acoplamiento existe. Un valor alto para CBO significa que las clases son estrechamente dependientes y, por lo tanto, es más probable que el hecho de cambiar una clase afecte a otras clases en el programa.
Respuesta por clase (*response for a class*, RFC)	RFC es una medida del número de métodos que potencialmente podrían ejecutarse en respuesta a un mensaje recibido por un objeto de dicha clase. Nuevamente, RFC se relaciona con la complejidad. Cuanto más alto sea el valor para RFC, más compleja será una clase y, por ende, es más probable que incluya errores.
Falta de cohesión en métodos (*lack of cohesion in methods*, LCOM)	LCOM se calcula al considerar pares de métodos en una clase. LCOM es la diferencia entre el número de pares de método sin compartir atributos y el número de pares de método con atributos compartidos. El valor de esta métrica se debate ampliamente y existe en muchas variaciones. No es claro si realmente agrega alguna información útil además de la proporcionada por otras métricas.

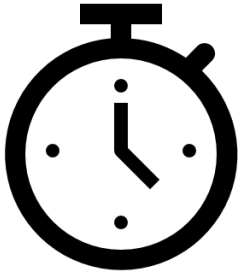
Métricas del producto - LDC – LÍNEAS DE CÓDIGO

La métrica más común para el tamaño

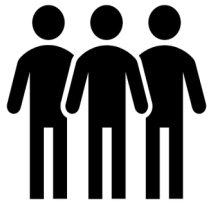
postmorten

Medida discutida !!

Resultados



¿Qué tiempo?



¿Cuántas personas?

Si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño

Utilidad de las métricas postmortem

- 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

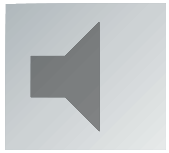
Métricas orientadas al tamaño

- Se puede obtener :

Productividad: relación entre KLDC / Persona mes

Calidad: relación entre Errores / KLDC

Costo: relación entre $ / KLDC



KLDC (miles de líneas de código)
LDC - LÍNEAS DE CÓDIGO

Como manejo las líneas

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

Productividad = KLDC/persona-mes
Calidad = errores/KLDC
Documentación = págs.. Doc./ KLDC
Costo = $/KLDC

- 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/Personas-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

- ¿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)?

Métrica de Punto función

PF- Punto función (AI

Factor de Ponderación, es subjetivo

Mide la cantidad de funcionalidad de un sistema descripto en una especificación

PF = TOTAL * [0.65 + 0.0
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]	=

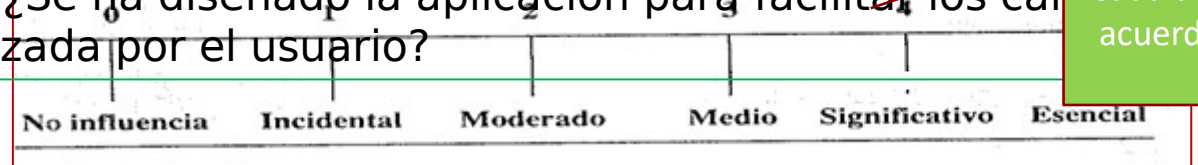
TOTAL

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

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 cam
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

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

»Pressman Cap. 15

Desarrollo de una métrica- GQM

- 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.

GQM (OPM)

- Estructura :

  Nivel **Conceptual** (Goal / Objetivo).

  Se define un objetivo (en nuestro caso, para el proyecto).

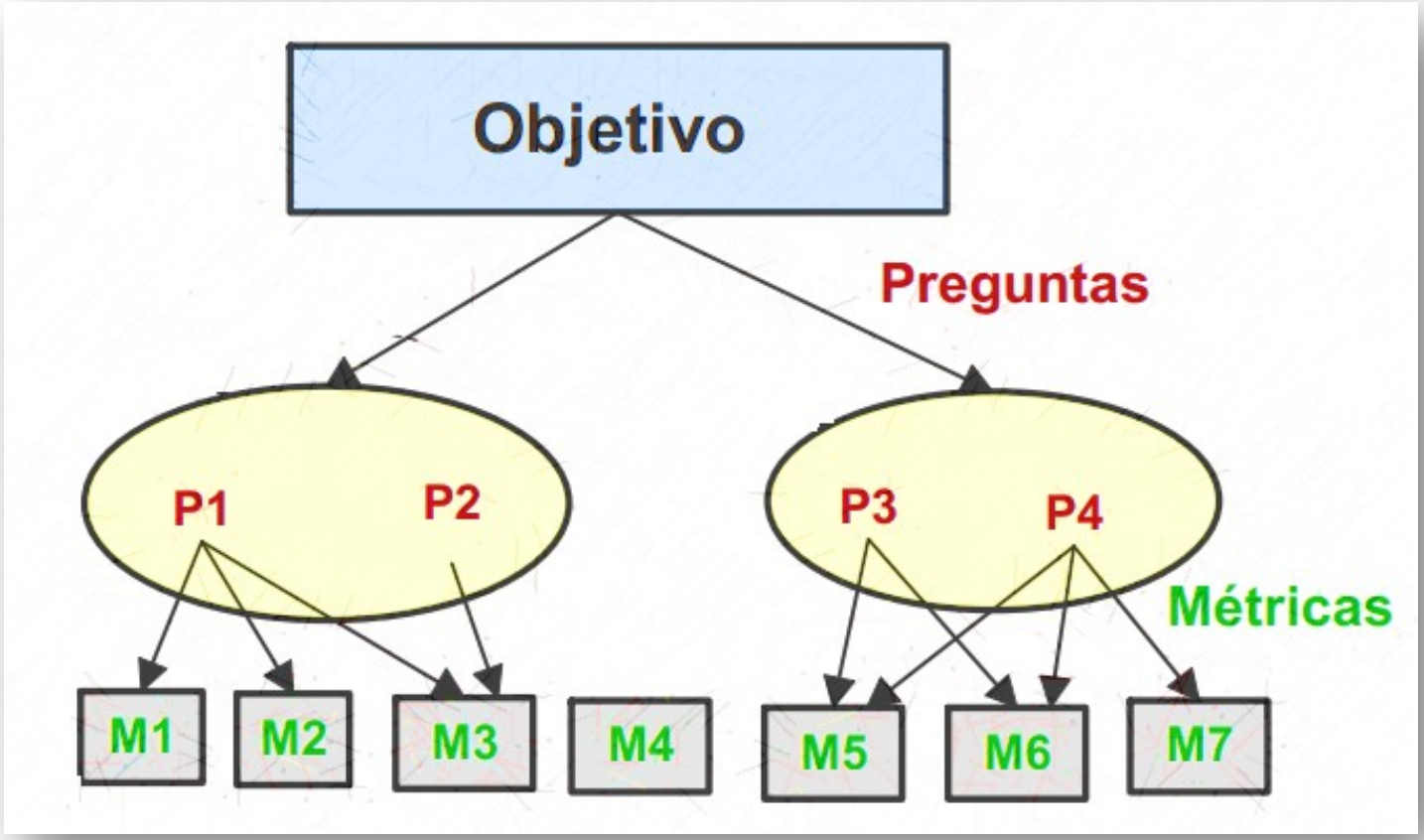
  Nivel **Operativo** (Question / Pregunta).

  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 métricas para cada pregunta, de modo de responder a cada una de un modo cuantitativo.

GQM (OPM)



GQM Ejemplo

- Evaluar ea Asigna

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

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

GQM (OPM)

- Es útil para decidir qué medir.
- Debe estar orientado a metas.
- Es flexible.

Gestión de Proyectos

Estimaciones

Estimaciones

- ¿Qué son?
- ¿Diferencias con las métricas?

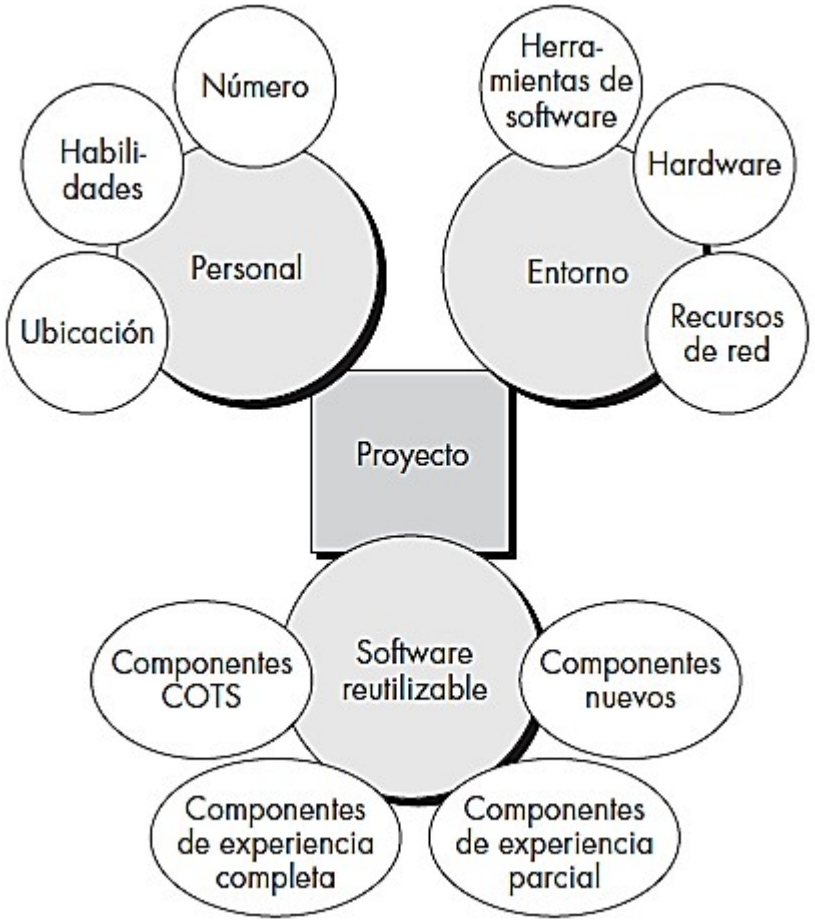
- ¿Cómo usarlas?

Estimaciones

- ¿Qué podemos estimar?

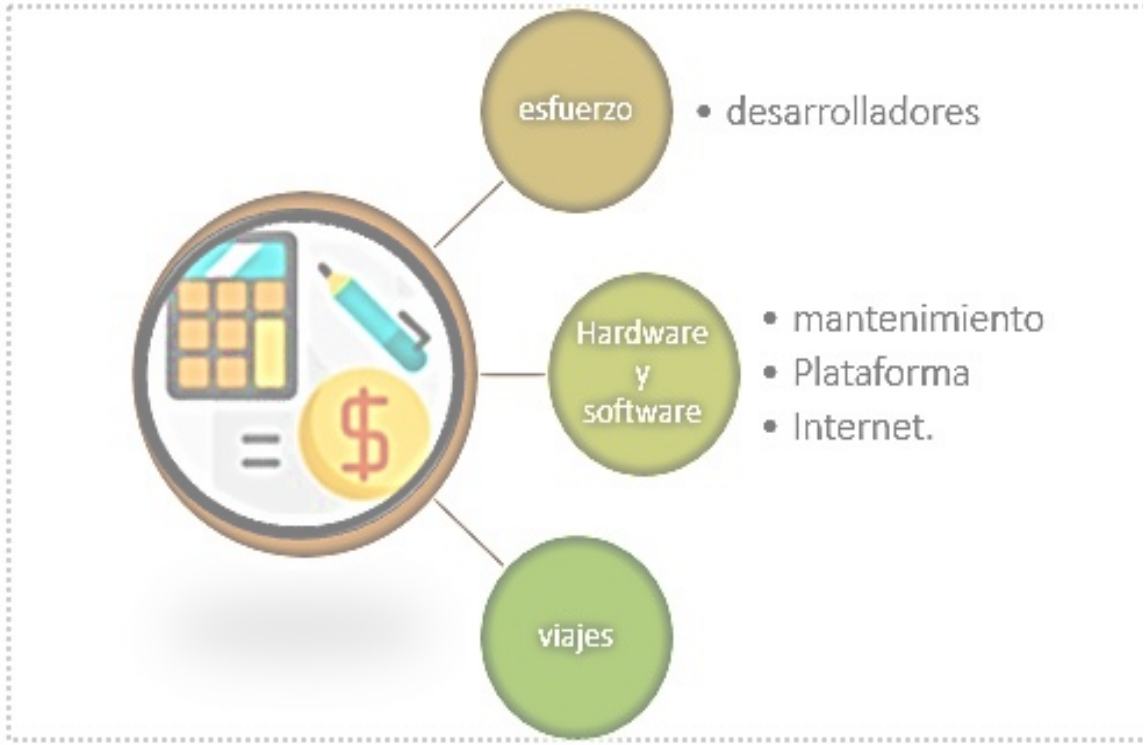
- ¿Qué tener en cuenta?
- ¿Qué factores influyen?

Estimaciones de recursos



Estimaciones de costos

3 parámetros para calcular costo de un proyecto



Fijación de precio - Relación Precio Costo

- ¿Qué tener en cuenta ?

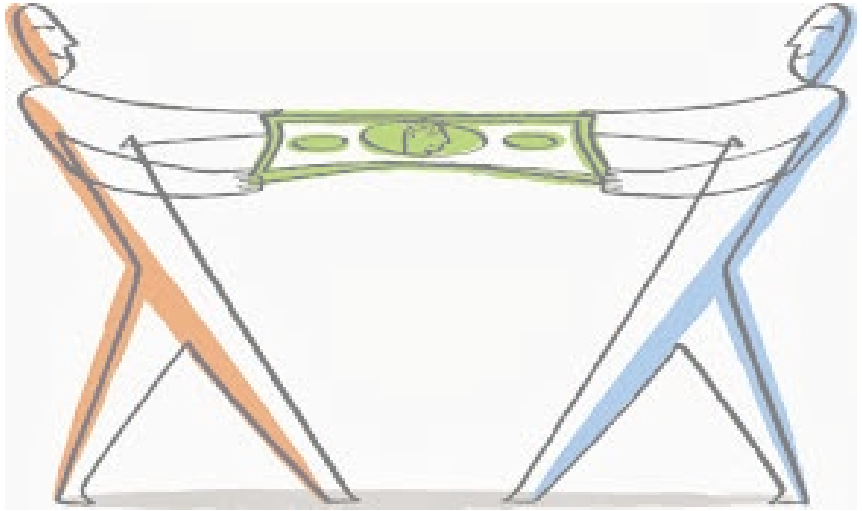
Oportunidad de mercado	Una organización de desarrollo podría ofertar un bajo precio debido a que desea conseguir cuota de mercado. Aceptar un beneficio bajo en un proyecto podría darle la oportunidad de obtener más beneficios posteriormente. La experiencia obtenida le permite desarrollar nuevos productos.
Incertidumbre en la estimación de costes	Si una organización está insegura de su coste estimado, puede incrementar su precio por encima del beneficio normal para cubrir alguna contingencia.
Términos contractuales	Un cliente puede estar dispuesto a permitir que el desarrollador retenga la propiedad del código fuente y que reutilice el código en otros proyectos. Por lo tanto, el precio podría ser menor que si el código fuente del software se le entregara al cliente.
Volatilidad de los requerimientos	Si es probable que los requerimientos cambien, una organización puede reducir los precios para ganar un contrato. Después de que el contrato se le asigne, se cargan precios altos a los cambios en los requerimientos.
Salud financiera	Los desarrolladores en dificultades financieras podrían bajar sus precios para obtener un contrato. Es mejor tener beneficios más bajos de los normales o incluso quebrar antes de quedar fuera de los negocios.

Fuente: Sommerville Cap 23

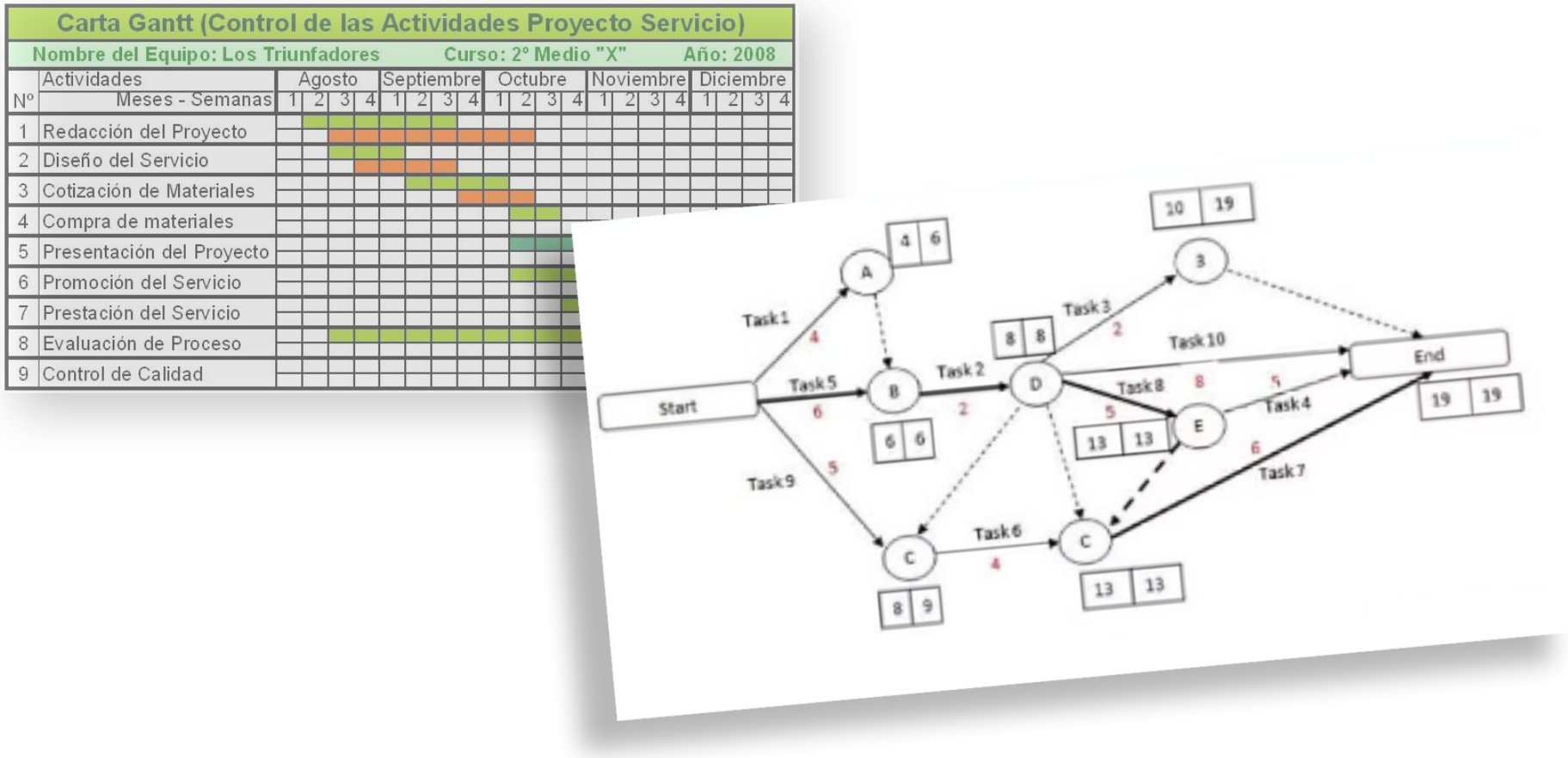
Fijación de precio

- Debe pensarse en :
- los intereses de la empresa,
- los riesgos,
- el tipo de contrato.

Esto puede hacer que el precio suba o baje.



Estimaciones de tiempo

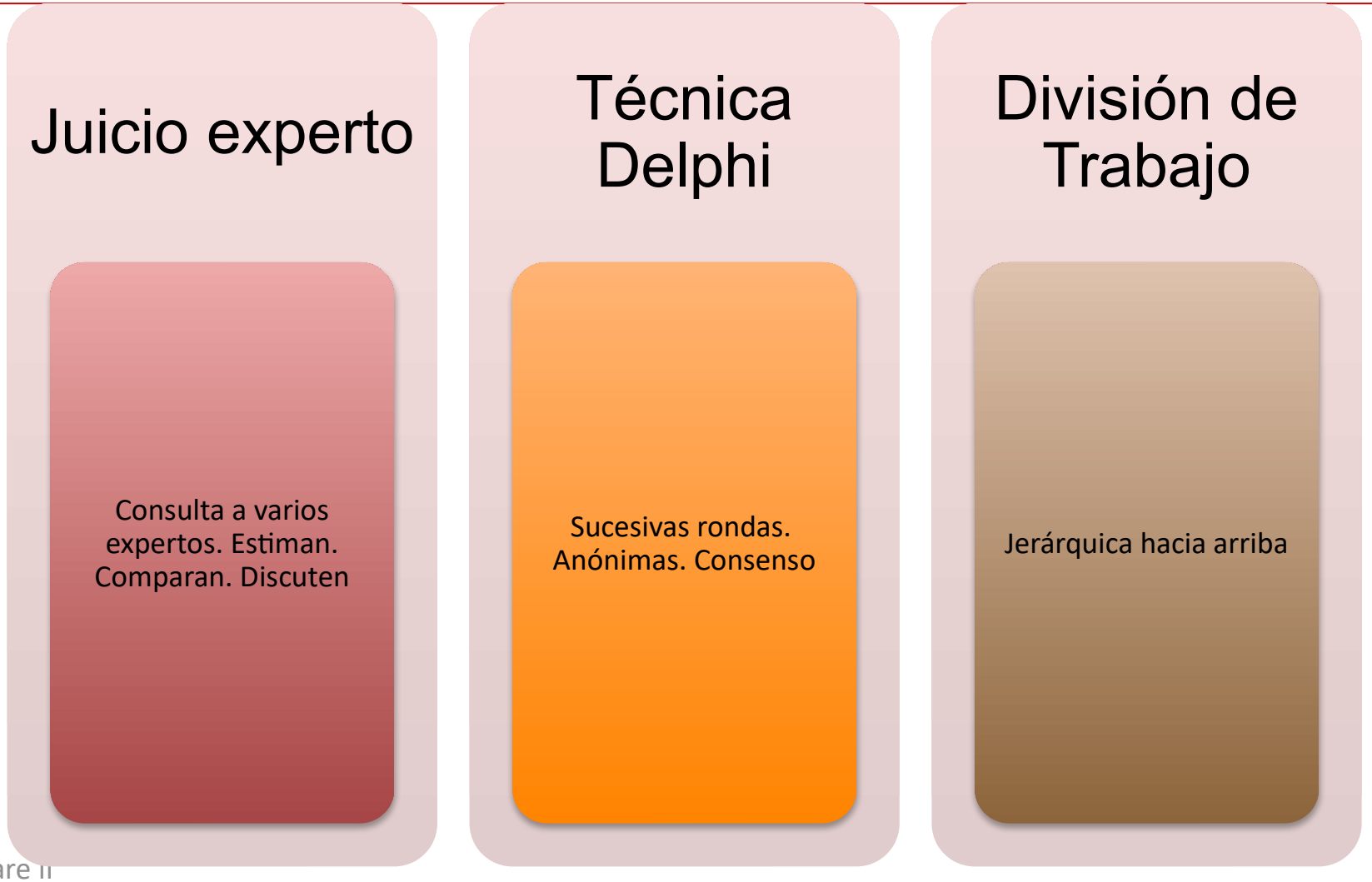


Carta Gantt (Control de las Actividades Proyecto Servicio)

Nombre del Equipo: Los Triunfadores Curso: 2° Medio "X" Año: 2008

Nº	Actividades / Meses - Semanas	Agosto 1	2	3	4	Septiembre 1	2	3	4	Octubre 1	2	3	4	Noviembre 1	2	3	4	Diciembre 1	2	3	4
1	Redacción del Proyecto																				
2	Diseño del Servicio																				
3	Cotización de Materiales																				
4	Compra de materiales																				
5	Presentación del Proyecto																				
6	Promoción del Servicio																				
7	Prestación del Servicio																				
8	Evaluación de Proceso																				
9	Control de Calidad																				

Técnicas de estimación



Técnicas de estimación- Juicio experto

Se utiliza a menudo para evaluar los insumos a utilizar. También es aplicado a todos los detalles técnicos y de gestión durante este proceso.

Es una forma útil de obtener una perspectiva externa del trabajo, pero es subjetiva y necesita pautas.

También se puede usar el juicio de expertos:

- Cuando hay mucho en juego. En un proyecto de alto impacto, será muy útil obtener una validación de expertos antes de avanzar.
- Antes de la planificación del proyecto.
- Para la gestión de costos. Para determinar costos para los precios de productos.
- Para la gestión de riesgos.

Técnicas de estimación- Planning Poker

- Participan todas las personas comprometidas en el desarrollo
- Cada una de las historias de usuario de un Sprint.
- Se llama asi porque se utilizan cartas numeradas. Lo normal es numerar las cartas con una serie de Fibonacci .

Pasos :

1. Se junta todo el equipo alrededor de una mesa. Cada integrante contará con un juego de cartas numeradas con la serie de Fibonacci.
2. El dueño de producto lee una historia de usuario a todo el equipo.
3. Si alguien tiene dudas se preguntan en ese momento.
4. Cada miembro del equipo, selecciona una carta y cuando todos han seleccionado una, se muestra.
5. Se comparan los resultados y se justifican. En caso de haber unanimidad se asigna el peso elegido a la historia. Caso contrario se vuelve a repetir el proceso hasta llegar a un consenso.
6. Se repite el proceso con cada historia de usuario en el product backlog.

Modelos empíricos de estimación

Utilizan fórmulas derivadas empíricamente para predecir costos o esfuerzo requerido en el proyecto

COCOMO II

Modelo empírico que derivó de recopilar datos a partir de un gran número de proyectos de software.

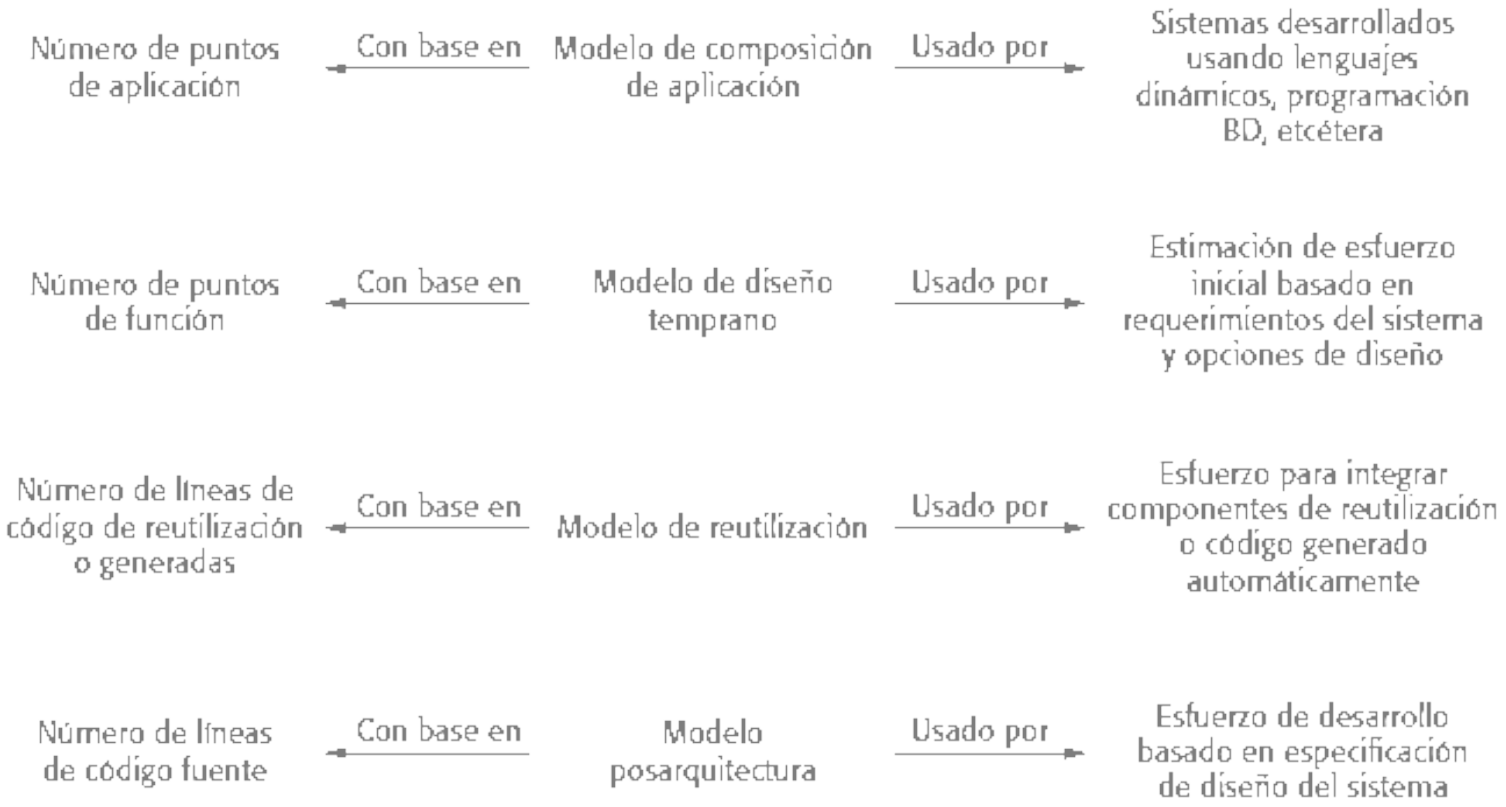
fórmulas  vinculan

el tamaño del sistema y los factores del producto, proyecto y equipo **CON**

el esfuerzo para desarrollar el sistema.

Fuente: Somerville Cap. 26

COCOMO II - Modelos



Estimaciones

- COCOMO 2:

http://www.cocomo2.com/