

5. Estimación en proyectos software

Índice

- Referencias
- Introducción
- Observaciones sobre la estimación
- Objetivos de la planificación del proyecto
- Actividades asociadas a la planificación
 - Ámbito del software.
 - Recursos.

Índice

- Estimación del proyecto del software
 - Técnicas de estimación.
- Técnicas de descomposición
 - Basadas en el problema.
 - LDC.
 - PF.
 - Basadas en el proceso.

Índice

- Modelos paramétricos de estimación
 - Introducción
 - Estructura común.
 - COCOMO II.
 - La ecuación del software.
 - Herramientas automáticas de estimación.
- Curvas de Rayleigh

Índice

- La decisión desarrollar-comprar
 - Introducción.
 - Subcontratación.
 - Determinación del precio de un proyecto.
- Conclusiones

Referencias

- Pressman, R.S. *Ingeniería del Software. Un Enfoque Práctico. Sexta Edición.* McGraw-Hill, 2005
- Sommerville, I. *Ingeniería del Software. 7ª edición.* Addison-Wesley, 2005
- COCOMO II Model Definition Manual,
http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html

Introducción

- El proceso de *gestión del proyecto de software* comienza con un conjunto de actividades que, globalmente se denominan *planificación* del proyecto
- La primera de estas actividades es la *estimación* de costes y tiempos

Introducción

- Como la estimación es la base de la planificación, hay que prestarle especial atención
- Como diría *Suntzu* en *El arte de la guerra*, siglo IV a.c.: "... si las estimaciones efectuadas antes de las hostilidades indican victoria, es porque los cálculos muestran que la fortaleza propia es superior a la del enemigo; si indican derrota, es porque los cálculos muestran que es inferior. Con mucho cálculos se puede ganar; con pocos, no. ¡Cuántas menos posibilidades de victoria tiene quien no hace ninguno!

Observaciones sobre la estimación

- La estimación la lleva a cabo el gestor del proyecto
- La estimación y planificación temporal de un proyecto software requiere:
 - Experiencia.
 - Buena información histórica.
 - Coraje de confiar en las métricas y la experiencia.

Observaciones sobre la estimación

- Hay diversos factores que influyen significativamente en las estimaciones:
 - La complejidad del proyecto.
 - El tamaño del proyecto.
 - Estabilidad de los requisitos.
 - Facilidad de identificar funciones.
 - Estructura de la información.
 - Disponibilidad de información histórica.

Objetivos de la planificación

- El *objetivo* de la planificación del proyecto de software es proporcionar un marco de trabajo que permita al gestor hacer estimaciones razonables de recursos, coste y planificación temporal
- Estas estimaciones se hacen al comienzo del proyecto
- Hay que actualizarlas según progresa éste

Objetivos de la estimación

- Además, deberían definirse los escenarios del *mejor caso* y del *peor caso* de forma que los resultados del proyecto puedan limitarse
- ¿Qué es el *peor caso* en un proyecto *real* de ingeniería? El túnel del Canal de la Mancha.

Actividades asociadas ... Ámbito del software

- La primera actividad de la planificación del proyecto es determinar el *ámbito del software*
- Recordemos que ámbito:
 - Contexto.
 - Objetivos de información.
 - Funciones y rendimiento.

Actividades asociadas ... Ámbito del software

- El cliente es el único que puede ayudarnos a determinar el ámbito
- Por tanto la comunicación con el cliente es fundamental
- La comunicación se puede iniciar con las *preguntas de contexto libre*
- Hay tres grupos dentro de estas preguntas

Actividades asociadas ... Ámbito del software

- El primer grupo se centra en el cliente, los objetivos globales y los beneficios:
 - ¿Quién está detrás de la solicitud de este trabajo?
 - ¿Quién utilizará esta solución?
 - ¿Cuál será el beneficio económico de una buena solución?
 - ¿Hay otro camino para la solución?

Actividades asociadas ... Ámbito del software

- El segundo grupo permiten comprender mejor el problema y que el cliente exprese sus percepciones sobre una solución:
 - ¿Cómo caracterizaría (el cliente) un resultado *correcto* que se generaría con una solución satisfactoria?
 - ¿Con qué problemas se enfrentará esta solución?
 - ¿Puede mostrarme (o describirme) el entorno en el que se utilizará la solución?

Actividades asociadas ... Ámbito del software

- ¿Hay aspectos o limitaciones especiales de rendimiento que afecten a la forma en que se aborde la solución?
- El último grupo se centra en la *efectividad* de la reunión. Se denominan *metacuestiones*:
 - ¿Es usted la persona apropiada para responder a estas preguntas? ¿Son *oficiales* sus respuestas?
 - Son relevantes mis preguntas para su problema.

Actividades asociadas ... Ámbito del software

- ¿Estoy realizando muchas preguntas?
- ¿Hay alguien más que pueda proporcionar información adicional?
- ¿Hay algo más que debiera preguntarle?
- Estas preguntas y otras similares ayudan a romper el hielo y a iniciar la comunicación esencial para establecer el ámbito del proyecto

Actividades asociadas ... Ámbito del software

- Sin embargo esto es el comienzo, y solo sirve para el primer encuentro
- Después deberían utilizarse técnicas de especificación más concretas.
- Por ejemplo disponemos de las *Técnicas Útiles para la Especificación de aplicaciones TUE* (FAST: *Facilitated Application Specification Techniques*)

Actividades asociadas ... Ámbito del software

- Las TUE presentan variaciones de la siguiente aproximación:
 - Se realiza la reunión en un lugar a convenir por clientes y desarrolladores.
 - Se propone una agenda lo suficientemente formal para cubrir todos los puntos importantes, pero lo suficientemente informal para permitir el flujo de ideas.

Actividades asociadas ... Ámbito del software

- Un moderador (desarrollador o cliente) controla la reunión.
- Se establece un mecanismo de presentación (transparencias, pizarra, etc.).
- Los objetivos son:
 - Identificar el problema.
 - Proponer elementos de la solución.
 - Negociar las distintas aproximaciones.
 - Especificar un conjunto preliminar de requisitos.

Actividades asociadas ... Ámbito del software

- Hay una *preparación* antes de la primera reunión TUE:
 - Antes de la reunión se debe realizar una entrevista en la que se realizan preguntas de contexto libre para ayudar a establecer el ámbito.
 - De esta reunión salen una o dos páginas de *requisitos del producto*.
 - Se eligen lugar, fecha, participantes y moderador de la reunión TUE y se les da estos requisitos con antelación.

Actividades asociadas ... Ámbito del software

- Antes de la reunión los asistentes deben redactar:
 - Una lista de objetos que son parte del entorno que rodea al sistema (e.g. archivos, dispositivos, etc.), que serán producidos por el sistema y que utiliza el sistema para llevar a cabo su funcionalidad.
 - Una lista de operaciones que manipulan o interactúan con los objetos.
 - Una lista de restricciones (e.g. coste, tamaño, etc.) y criterios de prestaciones (e.g. velocidad, precisión, etc.).

Actividades asociadas ... Ámbito del software

- Durante el desarrollo de la reunión TUE:
 - La primera cuestión que se debe tratar es la *necesidad* y *razón* del nuevo producto.
 - Se presentan las listas utilizando el medio convenido de tal forma que se puedan modificar. No se permiten críticas ni debate.
 - Después de que cada persona expone su lista sobre un tema determinado, se realiza una lista aditiva combinada para cada tema (objetos, operaciones y restricciones).

Actividades asociadas ... Ámbito del software

- Una vez completadas las listas:
 - Se forman subequipos que realizan una *miniespecificación* para una o varias entradas de la lista.
 - Aquí se añaden o eliminan ideas y se permite la discusión.
- Por lo general, serán necesarias varias reuniones TUE

Actividades asociadas ... Ámbito del software

- Veamos un ejemplo de ámbito
 - Se va a desarrollar un sistema automático de gestión de pasos a nivel.
 - El sistema identifica al tren cuando se encuentra a 1 km del paso a nivel.
 - Una vez identificado se enciende una luz roja, suena una campana y se baja la barrera.
 - La señal debe bajarse en 30 sg. (el tren llega en 72 sg., ya que circula a 50 km/h).

Actividades asociadas ... Ámbito del software

- Después de que el último vagón del tren pase a 10 m. del paso a nivel, se levanta la barrera, deja de sonar la campana, y cuando la barrera está elevada se apaga la luz roja.
- El planificador del proyecto examina la especificación del ámbito y extra todas las funciones principales del software, así como el rendimiento y las restricciones.

Actividades asociadas ... Ámbito del software

| Función | Rendimiento | Restricciones |
|-----------------------|-----------------------|--------------------|
| Detectar tren | | 1 km. |
| Detectar último vagón | | 10 m. después paso |
| Bajar barrera | 30 sg. | |
| Subir barrera | lo más rápido posible | |
| Encender luz | 1 sg. | |
| Apagar luz | 1 sg. | |
| Sonar campana | 1 sg. | |
| Para campana | 1 sg. | |

Actividades asociadas ... Ámbito del software

- La función no es independiente del rendimiento y/o las restricciones
 - e.g. no es lo mismo bajar la barrera en 30 sg. que en 15 sg.
 - e.g. no es lo mismo detectar el último vagón que detectar al tren cuando pase a 500 m. después del paso.

Actividades asociadas ... Ámbito del software

- Otro aspecto importante durante la estimación es el de la *fiabilidad* del software
- Nótese que se mide a posteriori, pero estamos en estimación
- En función del tipo de aplicación este factor se vuelve más o menos crítico (e.g. videojuego vs. control aviones)

Actividades asociadas ... Ámbito del software

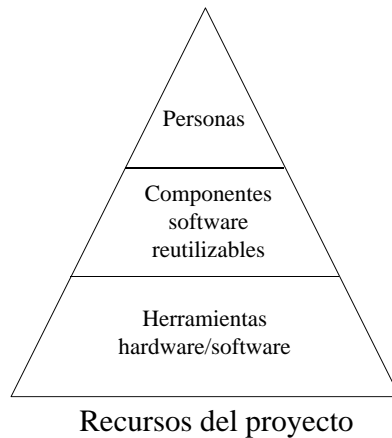
- Si se ha desarrollado la Especificación de Requisitos Software (SRS: *Software Requirements Specification*), el ámbito queda contenido en dicha SRS
- En otro caso, el gestor/planificador tendrá que *aproximarla* lo más posible mediante la declaración del ámbito.

Actividades asociadas ... Recursos

- La segunda tarea de la planificación del desarrollo de software es la *estimación de recursos* requeridos para acometer el esfuerzo de desarrollo
- Estos recursos pueden considerarse una pirámide:
 - Personas.
 - Componentes software reutilizables.
 - Herramientas de hardware/software.

Actividades asociadas ...

Recursos



Actividades asociadas ...

Recursos

- Cada recurso queda especificado mediante cuatro características:
 - Descripción del recurso.
 - Informe de disponibilidad.
 - Fecha cronológica en la que se requiere el recurso.
 - Tiempo durante el que será aplicado el recurso.
- Las dos últimas características son la *ventana temporal*

Actividades asociadas ...

Recursos

- Respecto al personal hay que especificar su posición en la organización y su especialidad
- El número de personas requerido debe estimarse como veremos en este tema

Actividades asociadas ...

Recursos

- Gran parte de la reutilización del software se debe a la tecnología de *componentes*
- Respecto a la planificación, tenemos cuatro tipos de componentes:
 - Ya desarrollados.
 - Ya experimentados.
 - Con experiencia parcial.
 - Nuevos.

Actividades asociadas ... Recursos

- Componentes ya desarrollados
 - Componente adquirido o existente.
 - Validados.
 - Listos para utilizarse.
- Componentes ya experimentados
 - Especificaciones, diseños, códigos o datos de prueba ya existentes desarrollados anteriormente y similares a los requeridos.

Actividades asociadas ... Recursos

- Miembros del equipo con experiencia completa en el dominio de aplicación de los componentes.
 - Riesgo bajo ante las modificaciones.
- Componentes con experiencia parcial
 - Especificaciones, diseños, códigos o datos de prueba ya existentes desarrollados anteriormente y relacionados con los requeridos.

Actividades asociadas ... Recursos

- Requieren una modificación sustancial.
 - Experiencia limitada del equipo en el dominio de aplicación de los componentes.
 - Alto riesgo ante las modificaciones.
- Componentes nuevos
 - Componentes a construir.

Actividades asociadas ... Recursos

- Directrices para la elección de componentes:
 1. Adquirir componentes ya desarrollados.
 2. Modificar componentes ya experimentados.
 3. No aceptar componentes con experiencia parcial

Actividades asociadas ... Recursos

- En cuanto a los recursos de entorno debemos constatar que no hay problemas con el software de desarrollo (e.g. número de licencias)
- También hay que planificar el acceso a hardware o recursos específicos (e.g. la barrera)

Estimación del proyecto de soft. Técnicas de estimación

- Estimar el coste del software es vital
- Las estimaciones *nunca* podrán ser exactas
- Cuanto mejor estimemos, más rentable será nuestro proyecto
- Estimar es difícil, ya que:
 - Los requisitos iniciales no están totalmente delimitados.
 - Puede que necesitemos utilizar tecnologías nuevas.

Estimación del proyecto de soft. Técnicas de estimación

- Las personas involucradas en el proyecto pueden tener distintos grados de experiencia.
- En cualquier caso, estimaremos
- Técnicas de estimación
 - *Retrasar* la estimación lo máximo posible. Cuanto más la retrasemos, más precisa será.
 - *Estimación por analogía*. Utilizar el coste de proyectos similares ya terminados.

Estimación del proyecto de soft. Técnicas de estimación

- *Ley de Parkinson*. El trabajo se extiende para rellenar el tiempo disponible.
- *Precio para ganar*. El coste se estima en todo el dinero que el cliente puede gastar en el proyecto.
- *Técnicas de descomposición*. Estiman el coste descomponiendo el producto y/o el proceso.
- *Modelos empíricos*. Modelos de regresión que relacionan esfuerzo con tamaño o funcionalidad

Técnicas de descomposición Basadas en el problema

- La técnica de *descomposición basada en el problema*, se basa en la descomposición del producto en funciones y estimar el tamaño del software
- Por tanto, la primera estimación que sirve de base para todas las demás, es la estimación del *tamaño* del software

Técnicas de descomposición Basadas en el problema

- Podemos considerar dos *tamaños del software*:
 - Tamaño en LDC.
 - Tamaño en PF.
- En cualquier caso, la *precisión de la estimación* depende de:
 - El grado en el que el planificador ha estimado adecuadamente el tamaño del producto a construir.

Técnicas de descomposición Basadas en el problema

- La habilidad para traducir la estimación del tamaño en esfuerzo y dinero. Depende fundamentalmente de la existencia de métricas.
- El grado en que el plan del proyecto refleja las habilidades del equipo de software.
- La estabilidad de los requisitos y del entorno que soporta el esfuerzo de la IS.

Técnicas de descomposición Basadas en el problema

- Nótese que las LDC-PF se utilizan de dos formas durante la estimación:
 - Variable de estimación para dimensionar al software.
 - Medida de línea base recopilada de proyectos anteriores que se utiliza junto a la variable de estimación para hacer la estimación.
- Además, las mediciones deberían estar clasificadas por dominios de proyectos

Técnicas de descomposición Basadas en el problema

- En el caso de que deseemos estimar las LDC debemos descomponer el problema llegando a un alto nivel de detalle
- Cuanto más grande sea la partición, más precisión tendrá nuestra estimación
- Para cada función estima un tamaño en LDC respecto a un valor *óptimo, medio y pesimista*

Técnicas de descomposición Basadas en el problema

- Dicha estimación puede basarse en:
 - Datos históricos.
 - Experiencia/intuición.
- Con estos valores se calcula un *valor esperado*:
$$VE = (V_o + 4V_m + V_p)/6$$
- Una vez estimado el tamaño se aplican los datos históricos de productividad LDC

Técnicas de descomposición Basadas en el problema

- e.g. supongamos que para un programa CAD tenemos identificado el siguiente ámbito (simplificando mucho):
 - El programa aceptará datos geométricos bi y tridimensionales.
 - Sobre los dibujos se podrá especificar una serie de transformaciones.
 - El sistema de archivo incluirá intercambio DXF.

Técnicas de descomposición Basadas en el problema

- El sistema puede disponer de varios periféricos, en particular, ratón, digitalizadora, uno o dos monitores y plotter.
- Se utilizará una interfaz basada en menús y ventanas.
- Supuesto un refinamiento (muy simplificado) podemos obtener los siguientes *estímulos*:

Técnicas de descomposición Basadas en el problema

- Módulo de dibujo (MD).
- Módulo de transformaciones (T).
- Módulo de archivos (A).
- Módulo de periféricos (MP).
- Interfaz (I).
- Siguiendo la técnica de estimación podemos desarrollar la siguiente tabla:

Técnicas de descomposición Basadas en el problema

| Función | Voptimista | Vmás probable | Vpesimista | VEsperado |
|---------|------------|---------------|------------|---------------------|
| MD | 8500 | 10500 | 13800 | 10717 |
| MT | 11000 | 15000 | 17200 | 14700 |
| MA | 6200 | 7700 | 8500 | 7583 |
| MP | 4000 | 5900 | 7400 | 5833 |
| I | 2700 | 3500 | 5000 | 3617 |
| | | | | total: 42450 |

Estimación del tamaño del software en LDC

Técnicas de descomposición Basadas en el problema

- Una revisión de datos históricos indica que la productividad *media* de la organización para este *tipo* de sistemas es de 650 (LDC/pm), y una tarifa laboral de 7800 (€/pm)

$$\text{esfuerzo} = 42450 \text{ (LDC)} / 650 \text{ (LDC/pm)} = 65 \text{ (pm)}$$

$$\text{coste} = 65 \text{ (pm)} * 7800 \text{ (€/pm)} = 507000 \text{ (€)}$$

Técnicas de descomposición Basadas en el problema

- Nótese que disponemos de una métrica de línea base sobre el coste (€/LDC)
- Entonces, ¿por qué recurrir a la tarifa laboral y no a la métrica?
- Puede deberse a que la métrica computa costes del proyecto no derivables exclusivamente del esfuerzo (e.g. hardware, viajes, formación, etc.)

Técnicas de descomposición Basadas en el problema

- También podemos estimar el tamaño del software en base a su funcionalidad (PF)
- En este caso, se parte de la descomposición del problema, pero no es necesario llegar al nivel de detalle que en el caso de LDC
- Ahora hay que estimar los parámetros de medición y los factores de ajuste de complejidad

Técnicas de descomposición Basadas en el problema

- Al igual que antes, esta estimación se basa en:
 - Datos históricos.
 - Experiencia/intuición.
- Con estos valores se calcula un *valor esperado*:

$$VE = (V_o + 4V_m + V_p)/6$$
- Una vez estimado el tamaño se aplican los datos históricos de productividad PF

Técnicas de descomposición Basadas en el problema

- Retomando el ejemplo del programa CAD

| parámetro medición | Voptimista | Vmás probable | Vpesimista | VEsperado | peso medio | |
|-------------------------|------------|------------------|------------|-----------|---------------|------------------|
| #entradas | 22 | 27 | 30 | 27 | 4 | 108 |
| #salidas | 15 | 20 | 29 | 21 | 5 | 105 |
| #peticiones | 17 | 25 | 28 | 23 | 4 | 92 |
| #archivos | 4 | 4 | 5 | 4 | 10 | 40 |
| #interfaces externas | 2 | 2 | 3 | 2 | 7 | 14 |
| | | | | | | total:359 |

Estimación del valor *total*

Técnicas de descomposición Basadas en el problema

F1: 4; F2: 2; F3: 0; F4: 4; F5: 4; F6: 5; F7: 4;
F8: 3; F9: 5; F10: 5; F11: 4; F12:3; F13: 5; F14: 5

$$PF = total * (0,65 + 0,01 * \sum_{i=1..14} F_i)$$

$$PF = 359 * (0,65 + 0,01 * 53) = 424 (PF)$$

Técnicas de descomposición Basadas en el problema

- Una revisión de datos históricos indica que la productividad *media* de la organización para este *tipo* de sistemas es de 6 (PF/pm), y una tarifa laboral de 7800 (€/pm)
 $\text{esfuerzo} = 424 \text{ (PF)} / 6 \text{ (PF/pm)} = 71 \text{ (pm)}$
 $\text{coste} = 71 \text{ (pm)} * 7800 \text{ (€/pm)} = 553800 \text{ (€)}$

Técnicas de descomposición Basadas en el proceso

- La técnica más común para estimar un proyecto es basar la estimación en el proceso que se va a utilizar
- Utilizando el proceso identificamos un conjunto pequeño de actividades de trabajo o tareas de trabajo y se estima el esfuerzo requerido para llevar a cabo cada tarea

Técnicas de descomposición Basadas en el proceso

- También comienza con la descomposición del problema
- Se genera la WBS y se calcula el esfuerzo para cada tarea de trabajo
- Retomando el ejemplo del programa CAD

Técnicas de descomposición Basadas en el proceso

| A.E. → | Com. cli. | Plan. | A. riesgo | Ingeniería | | Const. y adapta. | | Ev. cli. | esf. total |
|-------------------|-----------|-------|-----------|------------|--------|------------------|--------|----------------|----------------|
| acc. → | | | | Análisis | Diseño | Codific. | Prueba | | |
| función | | | | | | | | | |
| MD | | | | 1 | 6,5 | 2 | 5 | no esfuerzo | 14,5 |
| I | | | | 0,25 | 3 | 1 | 1,25 | | 5,5 |
| MT | | | | 2 | 9,5 | 3 | 6,5 | | 21 |
| MA | | | | 0,75 | 4,5 | 1,75 | 3,5 | | 10,5 |
| MP | | | | 0,25 | 4 | 0,75 | 3 | | 8 |
| esf. total | 0,5 | 0,5 | 0,5 | 4,25 | 27,5 | 8,5 | 19,25 | | 61(pm) |
| % esf. | 0,8% | 0,8% | 0,8% | 6,9% | 45,0% | 13,9% | 31,5% | | 100 (99,7)% |

Estimación por descomposición del proceso

Técnicas de descomposición Basadas en el proceso

- La tarifa laboral es de 7800 (€/pm)
 $\text{coste} = 61 \text{ (pm)} * 7800 \text{ (€/pm)} = 475800 \text{ (€)}$
- Nótese que hemos obtenido:
LDC: 65 (pm) PF: 71(pm) Proceso: 61 (pm)
Media: 65,7 (pm), variación máxima: 8% RAZONABLE
- Lo normal es obtener variaciones
razonables

Técnicas de descomposición Basadas en el proceso

- *Razonable* es en una franja del 20%
- Por encima no las estimaciones no son válidas
- Razones falta precisión:
 - No se entiende el ámbito o el planificador lo ha malinterpretado.
 - Datos históricos obsoletos o aplicados erróneamente.

Técnicas de descomposición Nota

- 1 persona trabajando todos los días (8h/d) durante un mes (22d):
 $e = 1 \text{ (p)} * 8 \text{ (h/d)} * 22 \text{ (d)} = 176 \text{ (ph)}$
Si queremos pasarlo a (pd):
 $176 \text{ (ph)} / 8 \text{ (h/d)} = 22 \text{ (pd)}$
Si queremos pasarlo a (pm):
 $22 \text{ (pd)} / 22 \text{ (d/m)} = 1 \text{ (pm)}$

Técnicas de descomposición Nota

- 1 persona trabajando todos los días a media jornada (4h/d) durante un mes (22d):
 $e = 1 \text{ (p)} * 4 \text{ (h/d)} * 22 \text{ (d)} = 88 \text{ (ph)}$
Si queremos pasarlo a (pd):
 $88 \text{ (ph)} / 8 \text{ (h/d)} = 11 \text{ (pd)}$
Si queremos pasarlo a (pm):
 $11 \text{ (pd)} / 22 \text{ (d/m)} = 0,5 \text{ (pm)}$

Modelos paramétricos de est. Introducción

- Utilizan formulas derivadas empíricamente para predecir el esfuerzo como una función de LDC o PF
- Estos valores de LDC o PF se obtienen por descomposición del problema
- Validez restringida al entorno donde se dedujo la fórmula

Modelos paramétricos de est. Estructura común

- La estructura global es:

$$E = A + Bx^C$$

donde:

E: esfuerzo (pm)

A, B, C: ctes. obtenidas empíricamente
x variable de estimación (LDC o PF)

Modelos paramétricos de est. Estructura común

- Modelos orientados a LDC:
 - Walston-Felix: $E = 5,2KLDC^{0,91}$
 - Bailey-Basisli: $E = 5,5 + 0,73KLDC^{1,16}$
 - Simple de Boehm: $E = 3,2KLDC^{1,05}$
 - Doty (KLDC>9): $E = 5,288KLDC^{1,0457}$

Modelos paramétricos de est. Estructura común

- Modelos orientados a PF:
 - Albretch y Gaffney: $E = -91,4 + 0,355PF$
 - Kemerer: $E = -37 + 0,96PF$
 - Proyectos pequeños: $E = -12,88 + 0,405PF$
- Nótese que para el mismo valor de LDC o PF los modelos difieren → necesitan calibración para las necesidades locales

Modelos paramétricos de est. COCOMO II

- COnstructive COst MOdel
- Boehm
 - COCOMO, 1981.
 - COCOMO II, 1996.
- COCOMO II define jerarquía de modelos:
 - Modelo de diseño previo.
 - Primeras etapas de IS.
 - Prototipado y evaluación.

Modelos paramétricos de est. COCOMO II

- Modelo de fase posterior a la arquitectura.
 - Requisitos establecidos.
 - Arquitectura básica del software establecida.
 - Construcción del software
- Modelos con una estructura común y una serie de parámetros que se pueden calibrar sobre una base de proyectos previos

Modelos paramétricos de est. La ecuación del software

- Putnam, 1992
$$E = B(LDC/P)^3(1/t^4)$$

donde

E: esfuerzo en personas-mes

t: duración del proyecto en años

B: factor especial de destrezas

P: parámetro de productividad

Modelos paramétricos de est. La ecuación del software

- Factor especial de destrezas
$$5 \leq KLDC \leq 15, B = 0,16$$
$$KLDC > 70, B = 0,39$$
- Parámetro de productividad refleja:
 - Madurez global del proceso y de las prácticas de gestión.
 - Amplitud corrección normas IS.
 - Nivel del lenguaje de programación.
 - Habilidad y experiencia del equipo de IS.
 - Complejidad de la aplicación.
 - e.g. sistemas empujados de tiempo real, $P = 2000$; software de telecomunicaciones y sistemas software, $P = 10000$; aplicaciones comerciales $P = 28000$.

Modelos paramétricos de est.

La ecuación del software

- La ecuación del software es paramétrica en:
 - LDC.
 - Duración del proyecto
- La ecuación del software simplificada:
 $t_{\min} = 8,14 (LDC/P)^{0,43} (m)$, $t_{\min} > 6$ meses
 $E = 180B(t_{\min}/12)^3 (pm)$, $E \geq 20$ personas-mes

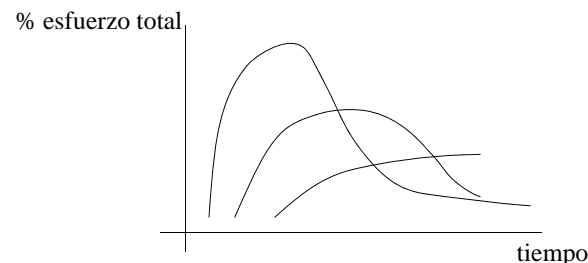
Modelos paramétricos de est.

Herramientas automáticas...

- Herramientas automáticas de estimación: implementan modelos paramétricos.
- Entrada:
 - Estimación en LDC o PF:
 - Características del proyecto.
- Salida:
 - Esfuerzo.
 - Costes.
 - Duración.

Curvas de Rayleigh

- Determinan que el esfuerzo en un proyecto no es constante en el tiempo.



Curvas de Rayleigh (aprox.)

La decisión desarrollar...

Introducción

- Muchas veces es más rentable adquirir el software de computadora que desarrollarlo
- Los gestores pueden optar por desarrollar el software o comprarlo
- Hay diversas *opciones* de adquisición:
 - Comprar ya desarrollado bajo licencia.
 - Adquirir componentes ya experimentados o parcialmente experimentados.

La decisión desarrollar...

Introducción

- Construirse de forma personalizada por una empresa externa para cumplir las especificaciones del comprador.
- Si es evidente que una opción es mucho más rentable que la otra se elige
- Si no, hacemos una comparación basada en:
 1. Desarrollo de una especificación del software.

La decisión desarrollar...

Introducción

2. Estimación del coste interno de desarrollo y de la fecha de entrega.
3. Selección de tres o cuatro aplicaciones que mejor cumplan las especificaciones.
4. Selección de componentes software que ayudarán en la construcción de la aplicación.
5. Desarrollo de una matriz de comparación.
6. Evaluación de cada paquete o componente.
7. Contacto con usuarios y petición de opiniones.

La decisión desarrollar...

Introducción

- En el análisis final se deben tener en cuenta las siguientes condiciones:
 - Fecha de entrega.
 - Coste total.
 - Coste de mantenimiento-soporte.

La decisión desarrollar...

Subcontratación

- Subcontratación (*outsourcing*): contratar las actividades de IS a un *tercero*, quien hace el trabajo a *bajo coste*, asegurando una *alta calidad*
- El trabajo de software de la *compañía origen* se reduce a la gestión de contratos

La decisión desarrollar... Subcontratación

- Puede ser una decisión estratégica o táctica
- Ventajas:
 - Ahorros de costes.
- Inconvenientes:
 - Pérdida del control del software.
 - Ponerse en manos de terceros.

La decisión desarrollar... Precio de un proyecto

- El coste de un proyecto software depende de tres parámetros:
 - Costes hardware y software, incluido su mantenimiento.
 - Costes de viajes y formación.
 - Costes de esfuerzo.

La decisión desarrollar... Precio de un proyecto

- En la mayoría de proyectos, el determinante, es el coste de esfuerzo, ya que:
 - El coste de hardware es *cerrado* y, hoy en día, no demasiado alto.
 - Aunque algunos proyectos tengan elevados costes de desplazamiento y/o formación, para la mayoría de proyectos son costes bajos.

La decisión desarrollar... Precio de un proyecto

- En las tarifas medias, no solamente se incluye el coste por esfuerzo
- También se incluyen los costes totales y se dividen por el número de programadores (y meses)
- Los siguientes costes también son parte del coste del esfuerzo total:
 - Coste de mantenimiento de la oficina.

La decisión desarrollar...

Precio de un proyecto

- Costes del personal no programador.
- Costes de red y comunicación.
- Costes de pensiones y/o seguros.
- Otros costes.
- Normalmente, el coste total es tres o cuatro veces el sueldo del programador:
tarifa programador: 2000(€/pm)
tarifa media: 7000(€/pm)

La decisión desarrollar...

Precio de un proyecto

- En base a esto podemos determinar:
$$\text{precio} = \text{coste} + \text{beneficios}$$
- Esta regla es muy sencilla
- Podemos ajustarla en base a los siguientes factores:
 - Cuota de mercado (alta/baja).
 - Estimación de costes poco clara.

La decisión desarrollar...

Precio de un proyecto

- Términos del contrato (propiedad del código).
- Volatilidad de los requisitos
(*honestos*/"*inteligentes*").
- Salud financiera.

Conclusiones

- Gestión → planificación → estimación
- Precisión estimación depende factores
- Actualizar planificación
- Casos mejor y peor
- Actividades: ámbito y recursos

Conclusiones

- Técnicas de estimación
 - Descomposición del problema.
 - LDC.
 - PF.
 - *Descomposición del proceso.*
 - Modelos paramétricos.
- Curvas de Rayleigh
- Desarrollar vs. comprar

Conclusiones

- Subcontratación
- Precio de un proyecto