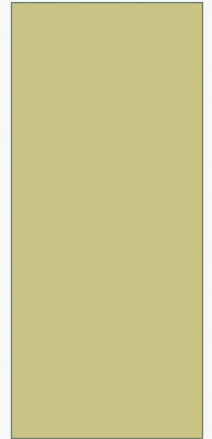


INGENIERÍA DEL SOFTWARE

UNIDAD 3: ESTIMACIÓN DE PROYECTOS DE SOFTWARE
CICLO LECTIVO 2013



OBJETIVOS DE LA CLASE

- Estimaciones seguras.
- Histórico de las estimaciones.
- Estimaciones seguras.
- Técnicas de descomposición.
- Tamaño del software.
- Estimación basada en el problema.
- Estimación basada en el proceso.
- Modelos empíricos de estimación-
- Desarrollar o comprar.
- Herramientas automáticas de estimación

ESTIMACIONES SEGURAS...

Recordemos de la anterior clase:

1. Dejar la estimación para más adelante .
2. Basar las estimaciones en proyectos similares ya terminados.
3. Utilizar **técnicas de descomposición** relativamente sencillas para generar las estimaciones de coste y de esfuerzo del proyecto.
4. Utilizar uno o más modelos empíricos para la estimación del coste y esfuerzo del software.

DEJAR LAS ESTIMACIONES PARA MAS ADELANTE...

- ¿Lo vemos como una opción factible?
¿Cómo organizamos un proyecto si no estimamos?
- Debemos realizar las estimaciones, sabiendo que contamos con pocos datos, y un alto grado de incertidumbre en muchos casos.

BASAR LAS ESTIMACIONES EN PROYECTOS SIMILARES

- Puede funcionar bien si el proyecto actual es bastante similar a los pasados.
- Normalmente, es muy difícil encontrar otro proyecto similar, no solo en las características básicas, sino en cuanto al cliente, al personal involucrado, etc.

ESTIMACIONES SEGURAS ..

- Las opciones restantes son métodos viables para la estimación del proyecto de software.
- Desde un punto de vista ideal, se deben aplicar conjuntamente las técnicas indicadas; usando cada una de ellas como comprobación de las otras.
- Las **técnicas de descomposición** utilizan un enfoque de **divide y vencerás** para la estimación del proyecto de software.
- Se pueden utilizar los **modelos empíricos** de estimación como complemento de las técnicas de descomposición, ofreciendo un enfoque de estimación potencialmente valioso por derecho propio.

ESTIMACIONES SEGURAS..

- Cada modelo se basa en la experiencia (datos históricos) y toma como base:

$$d = f(v_i)$$

- donde d es uno de los valores estimados (por ejemplo, esfuerzo, coste, duración del proyecto) y los v_i son determinados parámetros independientes (por ejemplo, LDC o PF estimados).

ESTIMACIONES SEGURAS

- Las **herramientas automáticas de estimación** implementan una o varias técnicas de descomposición o modelos empíricos.
- Cuando se combinan con una interfaz gráfica de usuario, las herramientas automáticas son una opción atractiva para la estimación. En sistemas de este tipo, se describen las características de la organización de desarrollo (por ejemplo, la experiencia, el entorno) y el software a desarrollar.
- De estos datos se obtienen las estimaciones de coste y de esfuerzo.

ESTIMACIONES SEGURAS ..

- Cada una de las opciones viables para la estimación de costes del software, sólo será buena si los datos históricos que se utilizan como base de la estimación son buenos. Si no existen datos históricos, la estimación del coste descansará sobre una base muy inestable.

TÉCNICAS DE DESCOMPOSICIÓN

La precisión de una estimación del proyecto de software se predice basándose en una serie de cosas:

1. El grado en el que el planificador ha estimado adecuadamente el tamaño del producto a construir;
2. la habilidad para traducir la estimación del tamaño en esfuerzo humano, tiempo y dinero (una función de la disponibilidad de métricas fiables de software de proyectos anteriores);
3. el grado en el que el plan del proyecto refleja las habilidades del equipo de software, y
4. la estabilidad de los requisitos del software y el entorno que soporta el esfuerzo de la ingeniería del software.

TAMAÑO DEL SOFTWARE

- El tamaño representa el primer reto importante del planificador de proyectos.
- En el contexto de la planificación de proyectos, el tamaño se refiere a una producción cuantificable del proyecto de software.
- Si se toma un enfoque directo, el tamaño se puede medir en LDC.
- Si se selecciona un enfoque indirecto, el tamaño se representa como PF.

TAMAÑO DEL SOFTWARE

Putnam y Myers sugieren cuatro enfoques diferentes del problema del tamaño:

- **Tamaño en lógica difusa.** Este enfoque utiliza las técnicas aproximadas de razonamiento que son la piedra angular de la lógica difusa. Para aplicar este enfoque, el planificador debe identificar el tipo de aplicación, establecer su magnitud en una escala cuantitativa y refinar la magnitud dentro del rango original.
- **Tamaño en punto de función.** El planificador desarrolla estimaciones de características del dominio de información

TAMAÑO DEL SOFTWARE

- **Tamaño de componentes estándar.** El software se compone de un número de componentes estándar que son genéricos para un área en particular de la aplicación. Por ejemplo, los componentes estándar para un sistema de información son: subsistemas, módulos, pantallas, informes, programas interactivos, programas por lotes, archivos, LDC e instrucciones para objetos. El planificador de proyectos estima el número de incidencias de cada uno de los componentes estándar, y utiliza datos de proyectos históricos para determinar el tamaño de entrega por componente estándar.

TAMAÑO DEL SOFTWARE

- **Tamaño del cambio.** Este enfoque se utiliza cuando un proyecto comprende la utilización de software existente que se debe modificar de alguna manera como parte de un proyecto. El planificador estima el número y tipo (por ejemplo: reutilización, añadir código, cambiar código, suprimir código) de modificaciones que se deben llevar a cabo. Mediante una proporción de esfuerzo para cada tipo de cambio, se puede estimar el tamaño del cambio.

ESTIMACIÓN BASADA EN EL PROBLEMA

Los datos de LDC y PF se utilizan de dos formas durante la estimación del proyecto de software:

1. como una variable de estimación que se utiliza para **dimensionar** cada elemento del software, y
2. Como métricas de línea base recopiladas de proyectos anteriores y utilizadas junto con variables de estimación para desarrollar proyecciones de coste y de esfuerzo.

ESTIMACIÓN BASADA EN EL PROBLEMA

- Las métricas de productividad de línea base (por ejemplo: LDC/pm o PF/pm) se aplican entonces para la variable de estimación adecuada y se extrae el coste o el esfuerzo de la función.
- Las estimaciones de función se combinan para producir una estimación global del proyecto entero.

ESTIMACIÓN BASADA EN EL PROBLEMA

- Los proyectos se deberían agrupar por tamaño de equipo, área de aplicación, complejidad y otros parámetros relevantes. Entonces se deberían calcular las medias del dominio local.
- Cuando se estima un proyecto nuevo, primero se debería asignar a un dominio, y a continuación utilizar la media del dominio adecuado para la productividad al generar la estimación.
- Las técnicas de estimación de LDC y PF difieren en el nivel de detalle que se requiere para la descomposición y el objetivo de la partición.

ESTIMACIÓN BASADA EN EL PROBLEMA

- Con independencia de la variable de estimación que se utilice, el planificador del proyecto comienza por estimar un rango de valores para cada función o valor del dominio de información.
- Mediante los datos históricos o (cuando todo lo demás falla) la intuición, el planificador estima un valor de tamaño optimista, más probable y pesimista para cada función, o cuenta para cada valor del dominio de información. Al especificar un rango de valores, se proporciona una indicación implícita del grado de incertidumbre.

VALOR ESPERADO PARA EL TAMAÑO DEL SOFTWARE

Entonces se calcula un valor de tres puntos o esperado.

- El valor esperado de la variable de estimación (tamaño), VE, se puede calcular como una media ponderada de las estimaciones optimistas (S_{opt}), las más probables (S_m), y las pesimistas (S_{pess}).

$$VE = (S_{opt} + 4 S_m + S_{pess}) / 6$$

- Se asume que existe una probabilidad muy pequeña en donde el resultado del tamaño real quedará fuera de los valores pesimistas y optimista

VALOR ESPERADO PARA EL TAMAÑO DEL SOFTWARE

- Una vez que se ha determinado el valor esperado de la variable de estimación, se aplican datos históricos de productividad de LDC y PF.
- ¿Son correctas las estimaciones?
- La única respuesta razonable a esto es: **No podemos estar seguros**. Cualquier técnica de estimación, no importa lo sofisticada que sea, se debe volver a comprobar con otro enfoque. Incluso entonces, va a prevalecer el sentido común y la experiencia.

ESTIMACIÓN BASADA EN EL PROCESO

- La técnica más común para estimar un proyecto es basar la estimación en el proceso que se va a utilizar.
- Es decir, el proceso se descompone en un conjunto relativamente pequeño de actividades o tareas, y en el esfuerzo requerido para llevar a cabo la estimación de cada tarea.
- Al igual que las técnicas basadas en problemas, la estimación basada en el proceso comienza con un esbozo de las funciones del software obtenidas a partir del ámbito del proyecto. Para cada función se debe llevar a cabo una serie de actividades del proceso de software

ESTIMACIÓN BASADA EN EL PROCESO

- Si la estimación basada en el proceso se realiza independientemente de la estimación de LDC o PF, ahora tendremos dos o tres estimaciones del coste y del esfuerzo que se pueden comparar y evaluar.
- Si ambos tipos de estimaciones muestran una concordancia razonable, hay una buena razón para creer que las estimaciones son fiables.
- Si, por otro lado, los resultados de estas técnicas de descomposición muestran poca concordancia, se debe realizar más investigación y análisis.

ESTIMACIONES BASADAS EN EL PROCESO

Muchas divergencias entre estimaciones se deben a una de dos causas:

- No se entiende adecuadamente el ámbito del proyecto o el planificador lo ha mal interpretado.
- Los datos de productividad usados para técnicas de estimación basados en problemas no son apropiados para la aplicación, están obsoletos (ya no reflejan con precisión la organización de la ingeniería del software), o se han aplicado erróneamente.
- El planificador debe determinar la causa de la divergencia y conciliar las estimaciones.

MODELOS EMPÍRICOS DE ESTIMACIÓN

- Un modelo de estimación para el software de computadora utiliza fórmulas derivadas empíricamente para predecir el esfuerzo como una función de LDC o PF.
- Los datos empíricos que soportan la mayoría de los modelos de estimación se obtienen de una muestra limitada de proyectos. Por esta razón, el modelo de estimación no es adecuado para todas las clases de software y en todos los entornos de desarrollo. Por consiguiente, los resultados obtenidos de dichos modelos se deben utilizar con prudencia

ESTRUCTURA DE LOS MODELOS DE ESTIMACIÓN

- Un modelo común de estimación se extrae utilizando el análisis de regresión sobre los datos recopilados de proyectos de software anteriores.
- La estructura global de tales modelos adquiere la forma de:

$$E = A + B * (ev)^C$$

- donde A , B y C son constantes obtenidas empíricamente, E es el esfuerzo en personas-mes, y **ev** es la variable de estimación (de LDC o PF).

ESTRUCTURA DE LOS MODELOS DE ESTIMACIÓN

- La mayoría de los modelos de estimación tienen algún componente de ajuste del proyecto que permite ajustar E por otras características del proyecto (por ejemplo: complejidad del problema, experiencia del personal, entorno de desarrollo).
- Entre los muchos modelos de estimación orientados a LDC propuestos en la bibliografía se encuentran los siguientes:

$$E = 5,2 * (KLDC)^{0,91} \quad \text{Modelo de Walston-Felix}$$

$$E = 5,5 + 0,73 * (KLDC)^{1,16} \quad \text{Modelo de Bailey-Basili}$$

Y otros ...

ESTRUCTURA DE LOS MODELOS DE ESTIMACIÓN

- También se han propuesto los modelos orientados a PF. Entre estos modelos se incluyen:

$E = -13,39 + 0,0545 \cdot PF$ Modelo de Albretch y Gaffney

$E = 60,62 * 7,728 * 10^{-8} PF^3$ Modelo de Kemerer

$E = 585,7 + 15,12 PF$ Modelo de Matson, Bamett y Mellichamp

- Un rápido examen de los modelos listados anteriormente indica que cada uno producirá un resultado diferente para el mismo valor de LDC y PF.

COCOMO

- El modelo COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria.
- El modelo original ha evolucionado a un modelo de estimación más completo llamado COCOMO II.

http://csse.usc.edu/csse/research/COCOMOII/cocomo_downloads.htm

DESARROLLAR O COMPRAR ..

- En muchas áreas de aplicación del software, a menudo es más rentable adquirir el software de computadora que desarrollarlo. Los gestores de ingeniería del software se enfrentan con una decisión *desarrollar o comprar* que se puede complicar aún más con las opciones de adquisición:
 - el software se puede comprar (con licencia) ya desarrollado;
 - se pueden adquirir componentes de software totalmente experimentado o parcialmente experimentado.
 - el software puede ser construido de forma personalizada por una empresa externa para cumplir las especificaciones del comprador.

DESARROLLAR O COMPRAR ..

Para productos de software más caros, se pueden aplicar las directrices siguientes:

- Desarrollo de una especificación para la función y rendimiento del software deseado. Definición de las características medibles, siempre que sea posible.
- Estimación del coste interno de desarrollo y la fecha de entrega.
 - Selección de tres o cuatro aplicaciones candidatos que cumplan mejor las especificaciones.
 - Selección de componentes de software reutilizables que ayudarán en la construcción de la aplicación requerida.

DESARROLLAR O COMPRAR ..

- Desarrollo de una matriz de comparación que presente la comparación una a una de las funciones clave. Alternativamente, realizar el seguimiento de las pruebas de evaluación para comparar el software candidato.
- Evaluación de cada paquete de software o componente según la calidad de productos anteriores, soporte del vendedor, dirección del producto, reputación, etc.
- Contacto con otros usuarios de dicho software y petición de opiniones.

OUTSOURCING

- La decisión de contratar fuentes externas puede ser estratégica o táctica.
- En el nivel estratégico, los gestores tienen en consideración si una parte importante de todo el trabajo del software puede ser contratado a otros.
- En el nivel táctico, un jefe de proyecto determina si algunas partes o todo el proyecto es aconsejable realizarlo mediante subcontratación.

OUTSOURCING

- Como ventajas, los ahorros de coste se pueden lograr reduciendo el número de personas y las instalaciones (por ejemplo: computadoras, infraestructura) que los apoyan.
- Y como desventajas, una compañía pierde control sobre el software que necesita. Como el software es una tecnología que diferencia sus sistemas, servicios, y productos, una compañía corre el riesgo de poner su destino en manos de un tercero.

HERRAMIENTAS AUTOMÁTICAS DE ESTIMACIÓN

- Las técnicas de descomposición y los modelos empíricos de estimación descritos en las secciones anteriores se pueden implementar con software.
- Todos realizan las seis funciones genéricas mostradas a continuación:
 - Dimensionamiento de las entregas del proyecto. Se estima el tamaño de uno o más productos de software.
 - Selección de las actividades del proyecto. Se selecciona el marco de trabajo del proceso adecuado.
 - Predicción de los niveles de la plantilla. Se especifica el número de personas disponibles para realizar el trabajo.
 - Predicción del esfuerzo del software.
 - Predicción del coste del software.
 - Predicción de la planificación del software.

¿Dudas, consultas?

