

Reutilización del conocimiento del diseño de software. Consideración en la determinación del precio de oferta al cliente

José M^a. Torralba Martínez

Área de Ingeniería de Organización. Facultad y Escuela Técnica Superior de Informática Aplicada. Universidad Politécnica de Valencia. Campus de Camino de Vera. 46071 Valencia, jtorral@omp.upv.es

Resumen

Se presenta la característica de la reutilizabilidad potencial del diseño y del producto en proyectos de software a medida y adaptados, y la posibilidad de activar algunos de los gastos de desarrollo para constituir una inversión. Con este supuesto, se elabora un modelo simple, presentado gráficamente, de determinación del precio al cliente basado en los costes.

Palabras clave: Presupuesto de costes de proyectos de software, precio de proyectos de software, reutilización del conocimiento del diseño de software.

1. Introducción

La empresa informática de desarrollo de software trabaja (“produce” o desarrolla) principalmente por proyectos (Cuevas, 2002) y en éstos hay tres elementos (objetivos) principales que orientan su gestión (Sanchís y Torralba, 2002, 21) y que deben cumplirse de acuerdo con lo contratado: a) Plazos de entregas, b) Especificaciones funcionales y calidad, y c) Precio. Esta comunicación* se orienta hacia el último de dichos elementos: el precio de los proyectos de software, a medida o adaptable, teniendo en cuenta la característica de ser reutilizable el conocimiento del diseño de software, como se describe a continuación.

1.1 Objeto

Se presenta la característica de reutilizabilidad del software y su posible consideración en el cálculo de costes para la determinación del precio de oferta al cliente en los proyectos a medida y adaptados. Se elabora un modelo simple, presentado gráficamente, sobre la consideración de la expectativa referida de reutilizabilidad al determinar el precio.

2. La reutilizabilidad del software

El término “software” se refiere no solo al código sino también a los datos, y la documentación de todo tipo que le acompaña, todo lo cual interesa en este trabajo. Algunos autores (Sommerville, 2001; Pressman, 2001) consideran que las características del producto software condicionan diversos aspectos de la Gestión del Proyecto, por lo que seguidamente recogeremos algunas de estas características.

* Este trabajo se deriva de la participación de su autor en un proyecto de investigación financiado por CICYT con referencia DPI2002-04342-C05-05, titulado “Diseño de un sistema de ayuda a la toma de decisiones para la gestión de los procesos de la empresa”.

2.1. Características del software

Como características del software en comparación con el hardware refiere la bibliografía (Pressman, 1997, 8; y Pressman, 2001, 5) las siguientes: a) La calidad depende sobre todo de la fase de diseño, b) La relación entre cantidad de personal (esfuerzo) y cantidad de software desarrollado depende mucho del plazo disponible para el desarrollo, c) La mayor parte de los costes se localizan en la fase de ingeniería (diseño), d) El software no se estropea con el uso; pero con el mantenimiento se pueden introducir errores, e) El mantenimiento correctivo es más complejo: no hay “componentes” de repuesto, y f) Todavía gran parte del software se construye a medida; la utilización de componentes previamente existentes es algo que está empezando.

Sommerville (2001, 9 y 72) indica algunas características que hacen la gestión de proyectos de software particularmente difícil: a) El producto es intangible; lo que afecta a la visibilidad del progreso del desarrollo de software, necesitando los directores una documentación específica para seguir el proceso de desarrollo; b) No existen procesos de software estándar; no se tiene una comprensión clara de las relaciones entre el tipo de proceso y el tipo de producto y de problemas; c) Con frecuencia los proyectos grandes son únicos; la experiencia está limitada a efectos de anticipar problemas; además la experiencia se convierte en obsoleta rápidamente por los rápidos cambios tecnológicos

Otra característica que nos interesa y que consideramos que se presenta de una forma diferente en el software, que pueden afectar sensiblemente a la Gestión del Proyecto, y en particular al proceso de estrategia del precio de oferta al cliente, es la reutilizabilidad potencial del software, que se considera a continuación.

2.2. La característica de reutilizabilidad

La característica referida de reutilizabilidad, esto es que es susceptible de ser reutilizado, y que de hecho se reutiliza, es reconocida en la bibliografía generalizadamente (Pfleeger, 2001). También se utiliza frecuentemente la denominación de reusabilidad.

La reutilización no es solo del código, o programa codificado en un lenguaje de programación, sino también de otros outputs o resultados que se obtienen a lo largo del ciclo de vida del desarrollo de software, tanto en su fase de proyecto (o diseño entendido en sentido amplio) como en la de ejecución del proyecto para obtener el producto; tanto el proyecto como su ejecución se considera que forman parte del software. No nos referimos en este trabajo específicamente al desarrollo de software basado en componentes, aunque es un tema actual en la investigación y también en el sector productivo de desarrollo de software que ofrece unas expectativas muy favorables (Pfleeger, 2001).

2.2.1. Reutilización y efecto experiencia

Reutilizar el software (Pfleeger, 2001), no es únicamente aprovechar la experiencia y aprendizaje adquiridos al desarrollar software, por lo que el estudio y repercusiones del carácter reutilizable se aparta algo de los análisis basados exclusivamente en la curva de experiencia y aprendizaje (Doménech, 1987, 24)

La reutilizabilidad ¿puede ser equivalente a conocimiento, experiencia, o aprendizaje explícito o documentado? Se puede considerar que hay alguna posible diferencia entre estos

dos conceptos que puede verse en el siguiente esquema sobre el proceso del conocimiento, relativo a las etapas de: captación, tratamiento y utilización del conocimiento.

a) Saber como se hace software: Formación

b) Haber desarrollado software: Proporciona la experiencia, que junto a la formación incrementa el conocimiento tácito de las personas. En esta situación, desarrollar software una vez más, aprovecha el conocimiento tácito adquirido en los desarrollos anteriores.

c) Documentar como se debe hacer software, aprovechando la experiencia adquirida: Conocimiento explícito.

d) En esta nueva situación, desarrollar software una vez más, aprovecha el conocimiento tanto tácito como explícito que aparece en la documentación, con posible soporte de herramientas.

e) Empleo de partes de productos software obtenidos en proyectos anteriores, con la adaptación que corresponda: A esto es a lo que nos referimos con el término reutilización y parece que es un aspecto algo particular. En este último caso estamos empleando partes de software que fueron diseñadas y construidas para un determinado proyecto, y que también sirven para otros proyectos de software posteriores, con la adaptación necesaria.

2.2.2. Reutilizabilidad en el ciclo de vida del software

La característica de reutilizabilidad la vamos a considerar en relación con dos procesos del ciclo de vida; a) el producto código, y b) el proyecto. En relación con el producto código, se considera que es reutilizable, y esto es específico del producto software, no siendo posible, por ejemplo, en el hardware o en otros artefactos creados con la ejecución de los proyectos (equipos, edificios e instalaciones, hardware incorporado a un coche, etc.). En efecto, se puede reutilizar una parte del código o programa (que es una parte del producto software) elaborado en el marco de un proyecto cuando posteriormente se lleve a cabo otro proyecto; y esto no es posible hacerlo, salvo en casos muy particulares, en proyectos que no son de software (como edificaciones, máquinas, hardware, etc.). En esta línea va el desarrollo de software basado en componentes.

En cuanto a la reutilizabilidad del proyecto, la situación ya no se puede considerar específica del software, ya que tienen esta propiedad también algunos elementos de la fase de diseño o proyecto (pero no del producto resultado de la ejecución, como se ha indicado) de otros artefactos como los señalados. Por ejemplo, los planos de un pequeño puente sobre un río son reutilizables en un proyecto posterior que esté relacionado con el previo, y con más facilidad si están realizados con un soporte CAD; lo mismo puede decirse del proyecto de un Instituto público de Enseñanza Secundaria, etc.

2.2.3. Significado económico de la reutilizabilidad

Para determinar el coste hay que considerar que si un recurso económico no se consume en una sola utilización, entonces su valor se reparte entre todas las potenciales utilidades del mismo, lo que es una de las funciones asignadas a la amortización técnica de los inmovilizados. En esta situación la parte no utilizada constituye una inversión. En cuanto a la vida del software, aunque una de las características es que no se desgasta con el uso, como ya se ha dicho (Pressman, 2001, 5), ni con la reutilización añadiríamos, sin embargo sí que tiene una vida limitada por diversas razones: cambios tecnológicos, el deterioro debido a las actuaciones por mantenimiento, etc.

Vamos a considerar la reutilizabilidad desde dos perspectivas temporales: reutilizabilidad potencial futura y reutilización de software preexistente. a) Reutilizabilidad potencial futura y

la creación de una inversión o activo. Si al hacer un proyecto X pensamos que puede ser reutilizado en el futuro para otros proyectos Y, Z, V,..., entonces algunos gastos realizados con ocasión del proyecto X podrían activarse, por lo que constituirían una inversión (Torralba, 1997) y ser recuperados con los proyectos Y, Z, V,..., decisión que está afectada del riesgo correspondiente como ocurre en toda inversión. Desde un punto de vista contable financiero podría tratarse con gastos de Investigación, Desarrollo e Innovación. b) Reutilización de software preexistente y su efecto en menor coste del proyecto. Si en un proyecto M, estamos reutilizando software obtenido en proyectos anteriores, entonces la necesidad de recursos (esfuerzo) puede ser menor, como veremos en un apartado posterior al referir el modelo de estimación de coste.

2.2.4. El tratamiento de la reutilizabilidad en los modelos de estimación del esfuerzo de desarrollo de software

a) Modelo COCOMO II (1999). En los modelos para los estadios de Diseño inicial y Post-arquitectura (Boehm y otros, 1999; Cuevas, 2002; Hale, 2000, 45), consideran cinco Factores de escala, siendo uno de ellos el siguiente: Precedentes (que coloca en el 1º lugar), con un rango de 1,33, y una ponderación entre 1,24 y 6,2 (El rango mide la aportación máxima posible de cada factor al esfuerzo total). En el modelo de Post-arquitectura se consideran 17 multiplicadores de esfuerzo, agrupados en cuatro categorías: a) Producto, b) Plataforma, c) Personal, y d) Proyecto; algunos de los multiplicadores se refieren a la experiencia y reutilización, como se presenta a continuación y resume la tabla 1.

Tabla 1. Multiplicadores de esfuerzo relativos a reutilización y experiencia en el modelo COCOMO II

Multiplicador	Código	Subgrupo	Rango	Ponderación	Nº. de orden del multiplicador
Reutilización requerida	RUSE	Producto	1,31	0,95-1,24	4
Experiencia desarrollando aplicaciones similares	AEXP	Personal	1,51	1,22-0,81	11
Experiencia con la plataforma de desarrollo	PEXP	Personal	1,40	1,19-0,85	12
Experiencia con el lenguaje y la herramienta	LEXP	Personal	1,43	1,20-0,84	13

La referencia que hace a la reutilización la califica de requerida, luego trata de medir el sobreesfuerzo que supone desarrollar software si se desea que sea reutilizable en el futuro, que es un planteamiento muy actual en el desarrollo de software. Nuestro punto de vista se centra sobre todo en que en general el software existente puede reutilizarse, aunque no se pensara en ello explícitamente cuando se desarrolló, luego este multiplicador no recoge la reutilización de un software preexistente que es lo que se está considerando en este trabajo.

La referencia a la experiencia es explícita, tanto como factor de escala (Precedentes) como a través de multiplicador de esfuerzo (experiencia desarrollando aplicaciones similares), con un rango combinado de 2,0083; si se incluyen también los otros tipos de experiencia (con la

plataforma de desarrollo; y con el lenguaje y la herramienta) el rango combinado es muy superior.

b) Modelo de IFPUG (International Function Point Users Group).- En cuanto a la reutilización, en el cálculo del tamaño, prevé dos enfoques (Montesa, 2003) : a) desarrollo pensando en la reutilizabilidad futura, y b) modificación de aplicaciones informáticas. El primero, en la línea de lo visto en el modelo COCOMO II, hace referencia al tamaño a considerar de una aplicación informática, de modo que se tienen en cuenta unas características generales del sistema en donde se evalúa la reutilizabilidad del software en otros software u adaptaciones futuras de éste.

Las características generales implicadas son (se mantiene el número de orden (IFPUG, 1999)):

- Diez. Reusabilidad, que se evalúa en los siguientes casos: a) Se reutilizará parte del software internamente, b) Se reutilizará parte del software por otros sistemas, c) Se desarrolla una aplicación con el objetivo de ser adaptada a otros menesteres, modificando parte del código en sí, y d) Se desarrolla una aplicación parametrizada, de modo que modificando algunos parámetros se utilizará por otros usuarios.
- Once. Facilidades de instalación. Este factor, está relacionado con la reutilización del trabajo realizado, cuando haya que instalar este software en otros lugares. Este factor contempla aspectos como: a) Se especifican características de la conversión e instalación, b) Se desea una instalación y conversión automáticas.
- Trece. Múltiples lugares. Hace referencia a que el software a desarrollar deberá utilizarse en diversos entornos.

El segundo enfoque indicado, que podemos asociar a modificaciones de aplicaciones informáticas, se le llama proyectos de incremento. En este tipo de proyectos se contempla que existe una aplicación previa a la que se va a añadir, modificar y eliminar funcionalidades. En este caso se considera tan solo el tamaño del proyecto, sin considerar el tamaño de la aplicación obtenida, que se medirá directamente. De este modo se calculan las funcionalidades siguientes: a) Puntos de función sin ajustar que aportan las funcionalidades a añadir (entradas, salidas, consultas, ficheros,..), b) Puntos de función sin ajustar que aportan las funcionalidades a modificar, c) Puntos de función sin ajustar que aportan las funcionalidades necesarias para realizar la conversión de datos, etc. que pudieran requerirse, y d) Puntos de función de la porción de la aplicación a ser suprimida (también dan trabajo). Por otra parte se contemplan las características generales de la aplicación tanto antes de la modificación, como después y se corrigen los puntos de función estimados previamente.

Se suman todos los puntos de función y se obtiene el tamaño del proyecto a desarrollar. Nótese que cuentan de no solo las funcionalidades añadidas, sino también las suprimidas, y esto es así porque suprimir funcionalidades en una aplicación, y hacerlo de forma correcta, puede ser tan laborioso como el desarrollo de otras funcionalidades.

En cuanto al cálculo del esfuerzo en sí, hay que partir de los datos históricos, es decir las productividades obtenidas en este tipo de aplicaciones, por supuesto que a medida que la empresa, como colectivo de desarrolladores, adquiera experiencia, podrá observar unas mejores productividades, consecuencia de la curva de la experiencia. En caso de que la empresa sea capaz de generar portafolios de diferentes tipos de aplicaciones, quedará mejor reflejada la experiencia en cada tipo de sistema. La forma de calcular la productividad parte del tamaño del portafolio (en puntos de función) es decir “Sumatorio de puntos de función de

las aplicaciones contenidas en el portafolio a ser medido” y el esfuerzo aplicado en el desarrollo de este portafolio. La productividad por punto de función se calcula dividiendo los puntos de función por el esfuerzo. Esta productividad variará dependiendo del tipo de aplicación, y sobre todo en aquellos casos en los que la empresa disponga de más experiencia.

3. Modelo de costes para determinar precio de oferta al cliente en proyectos de software a medida y adaptados

En la bibliografía de estrategia de precios (Kotler, 2003), se suelen considerar tres variables importantes que condicionan el precio: a) la competencia, b) la demanda y c) el coste (Díez, 2001), que también son aplicables a los proyectos de software con algunas adaptaciones (Torralba et al., 2001). Una de las variables básicas que condiciona el precio es pues el coste, y a él nos vamos a referir ahora a efectos de fijación del precio del software, aunque en la vida real habría que considerar también las otras variables.

Los modelos de coste de proyectos de software son mayoritariamente uniproyecto (Pfleeger, 2001), que también se podrían denominar unicliente, en relación al tema que consideramos. Esto es así porque consideran solo los costes en que se va a incurrir si se realiza el proyecto, correspondientes solo al cliente para el que se estima el presupuesto y la determinación del precio. Sin considerar los costes que se tendrá con clientes futuros en proyectos similares, que vienen muy condicionados por la reutilizabilidad.

Con el supuesto sobre reutilizabilidad, los modelos de costes deberían ser multiproyecto (Pfleeger, 2001), o multicliente, ya que deberían considerar que una parte del proyecto (y de su coste) se puede reutilizar en proyectos futuros y por lo tanto procedería tratar la parte a reutilizar como una inversión o activo que se puede recuperar en proyectos futuros, con otros clientes, razón por la cual en el modelo se deberían contemplar a los clientes potenciales futuros (multicliente) y a sus costes potenciales. Y esto es lo que planteamos en el modelo de determinación del precio considerando la expectativa de reutilización que presentamos.

3.1. Reutilizabilidad y estrategia de precio del software a medida y adaptado

El tratamiento de la reutilizabilidad como inversión, o como menor coste, que se han referido pueden ser considerados al fijar el precio así: a) Reutilizabilidad como inversión: se podría plantear no recuperar todos los gastos de un primer proyecto en el cliente que lo ha solicitado, y b) Reutilización como menor coste: los costes por unidad de funcionalidad asignables a los sucesivos clientes pueden decrecer por la reutilización.

Se presenta un modelo de determinación del precio de oferta al cliente, en proyectos-producto de software, que contempla la consideración como inversión de determinados gastos, como ya se ha dicho. Además se presenta también otro modelo previo que correspondería al supuesto de que no se considerara gasto alguno como inversión. Previamente, se presentan algunas simplificaciones y supuestos.

3.2. Simplificaciones generales del modelo

La elaboración de la oferta al cliente, en un mercado concurrencial y dinámico, es una actividad muy compleja, en la que intervienen planteamientos de política de precios y, también, de política general de empresa. Un proyecto-producto de software es complejo. Por todo ello, procede establecer algunos de los supuestos que explícitamente se consideran,

necesarios para una cierta representatividad del modelo elemental que se presenta. Estos supuestos son los siguientes:

A) Sobre terminología simplificada: a) Se refieren gastos y costes, por no entrar en precisiones de tipo contable.

B) En relación con el software que se desarrolla: a) Se trata de aplicaciones comunes, esto es, que no constituyen necesariamente una innovación, en ningún sentido; b) Se supone que no se producen cambios tecnológicos importantes. Estos pueden afectar, entre otros, a los siguientes aspectos: b-1) Cambios en las necesidades de recursos; b-2) El grado en que es reutilizable algún elemento del proyecto-producto; etc. Dada la rápida evolución tecnológica existente, este supuesto se cumpliría razonablemente en un plazo temporal acotado; c) Se mantienen algunos de los requerimientos de los clientes actuales en los proyectos futuros, a los que se les harán aplicaciones reutilizando las realizadas anteriormente.

C) En relación con el cálculo de costes y el precio: a) Se hace un planteamiento presupuestario. Las desviaciones que se produzcan respecto de los datos históricos darán lugar a modificación de los presupuestos, pero no se tienen en cuenta. De alguna forma, se considera que lo presupuestado coincide con lo histórico; b) No se ha considerado el desfase temporal de algunos gastos que se producen después de la entrega del software. Este es el caso de los gastos que origina el mantenimiento correctivo, si se supone que es por cuenta del proyectista-desarrollador; c) Se supone un nivel de inflación reducido. En caso contrario, habría que formular consideraciones adicionales sobre valores en términos reales que no se han tenido en cuenta; d) Los elementos económicos referidos están afectados de riesgo, ya que no se conocen con exactitud aspectos como los siguientes: d-1) Si se harán efectivos, en especial para los clientes 2º y siguientes, ya que se habla de reutilizabilidad potencial; d-2) Sus cuantías, caso de que se lleven a efecto; d-3) Las fechas de realización, etc.; e) El mercado admite la política de precios de la empresa. Como hay que respetar las condiciones del mercado, se supone que los precios ofertados por la empresa son compatibles con las condiciones del mercado (no se entra en los condicionantes de competencia y demanda).

Las figuras presentan las gráficas que representan los modelos: sin activar costes y con activación (reutilizando el conocimiento del proyecto y producto). En la figura 1, la evolución general de los costes de proyectos tipo debido a la experiencia y reutilizabilidad; las 2 y 3, el caso de que no se activen ciertos gastos por reutilizabilidad; las 4 y 5 el caso de activación y recuperación entre los dos primeros clientes, o los cuatro.

4. Comentario final

En base al modelo simple de costes presentado, se podrían elaborar modelos más completos que ayudarían en la determinación de precios de oferta al cliente en proyectos de software, a medida y adaptados, que tuvieran en cuenta esta característica específica del software de su carácter reutilizable.

Agradecimientos

A. D^a. Gloria I. Moncaleano y D. Manuel Rodenes, por su importante colaboración en este trabajo. Al Instituto Tecnológico de la Informática de la Comunidad Valenciana. A las Empresas que muy generosamente y de forma muy eficaz nos facilitaron la información propia en que se basa el trabajo realizado que sirve de base a esta comunicación. A D. Rafael Oliver, D^a. Laura Olcina y D. José Bernabeu. A D. Victor Ruíz y D. Pablo Boix, con los que

intercambio ideas sobre estos temas aprovechando su experiencia. D. Francisco Sanchís, D. José Manuel Santaúrsula, D. Julián Marcelo, D. Joaquín Ordieres, D. José Onofre Montesa. A D. Edwin Rodríguez, D. Salvador Pons Alberola y D. Jesús Marqués Parada.

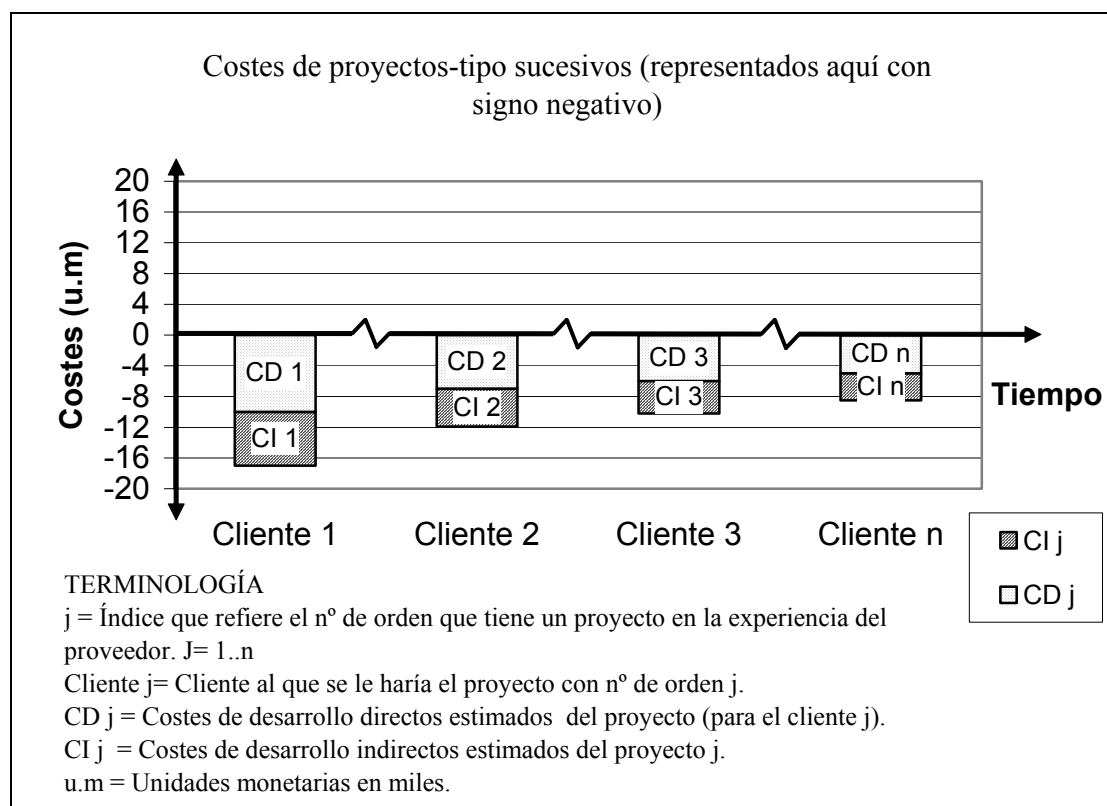


Figura 1. Evolución de los costes de proyectos tipo sucesivos. Fuente: Basado en [Torralba, 2003].

Referencias

- Cuevas, G. (2002). Gestión del proceso software. Centro de Estudios Ramón Areces.
- International Function Point Users Group (1999): *User Manual*. V. 4. IFPUG.
- Díez de Castro, E. C. (2001): Política de Precios. ESIC.
- Kotler, P. (2003): *Marketing*. McGraw-Hill.
- Nagle, J., Holden, H. (2002): Estrategia de Precios. Ed. McGraw-Hill.
- Pfleeger, S. L. (2001). *Software engineering. Theory and Practice*. Prentice Hall.
- Pressman, R. (2001 y 1998). *Software engineering*. McGraw-Hill.
- Sanchís, F. y Torralba, J. M^a. (2002). Proyectos de Ingeniería Informática. Servicio de Publicaciones de la EUI. UPM.
- Sommerville, J. (2001). *Software engineering*. Addison Wesley.
- Torralba, J. M^a. (1997): Consideración del presupuesto del proyecto-producto como una inversión para el proyectista-desarrollador en proyectos de software. *Actas del Congreso de Ingeniería de Proyectos*, Sevilla.
- Torralba, J. M^a. (2003): Introducción al presupuesto del proyecto informático y telemático. Servicio de Publicaciones UPV.
- Torralba, J. M^a., Chiner, M y Ferrer, R. (2001): Del presupuesto al precio de oferta en proyectos informáticos. Bonet.
- Torralba, J. M^a. (2004): Economía y gestión de proyectos. Documento interno. UPV.

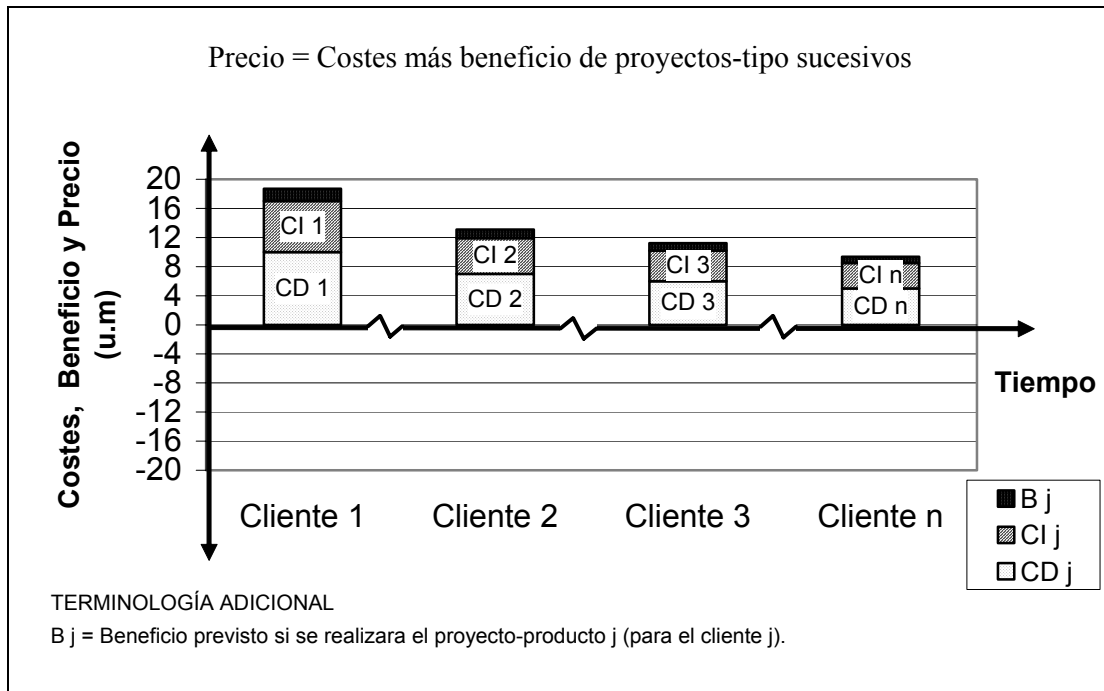


Figura 2. Costes más beneficio de proyectos-tipo sucesivos, si no se activan gastos por reutilizabilidad. Fuente: Basado en [Torralba, 2003].

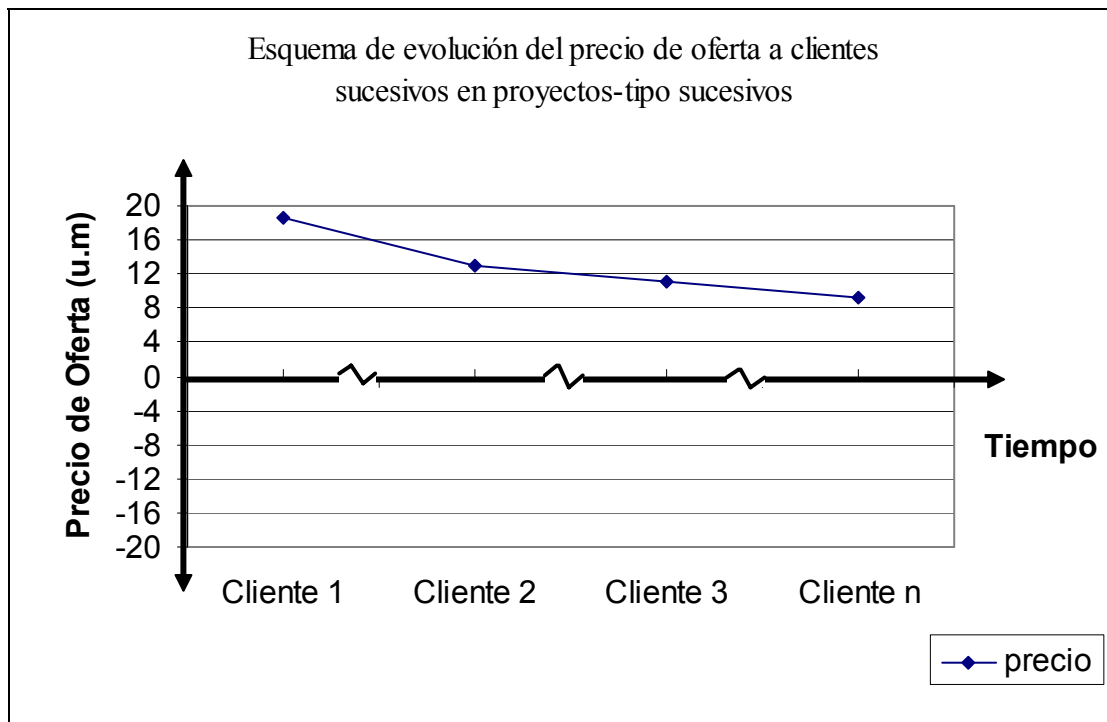


Figura 3. Esquema de evolución del precio de oferta al cliente de proyectos-tipo sucesivos, si no se activan gastos por reutilizabilidad. Fuente: Basado en [Torralba, 2003].

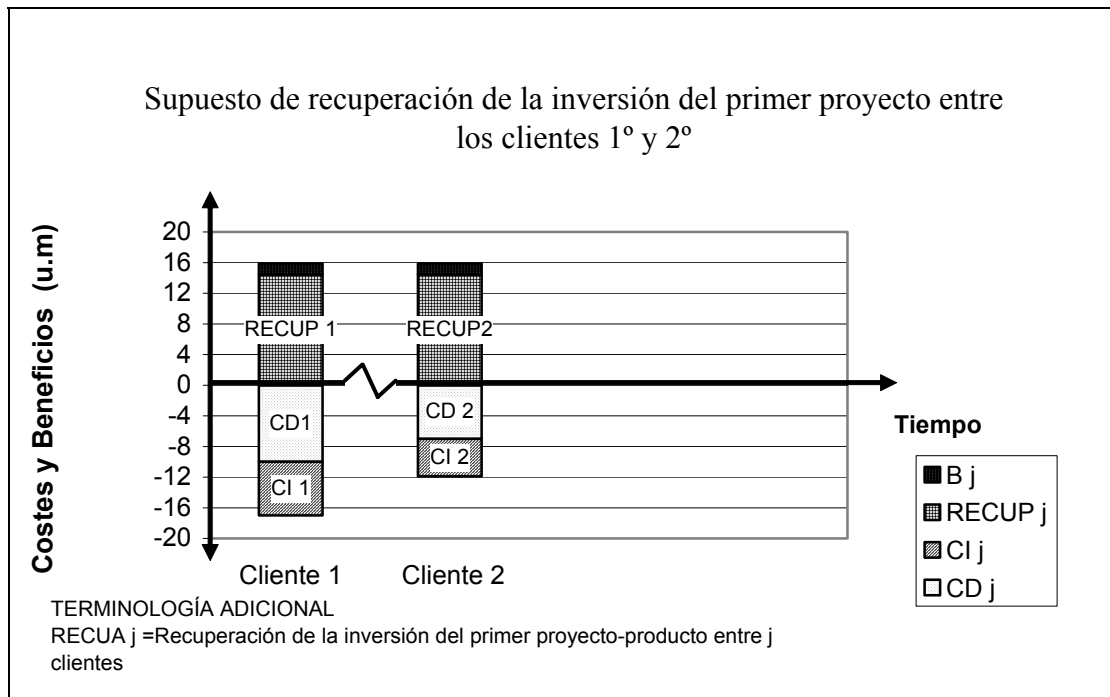


Figura 4. Supuesto de recuperación de la inversión del primer proyecto entre los clientes 1º y 2º (planteamiento estático). Fuente: Basado en [Torralba, 2003]

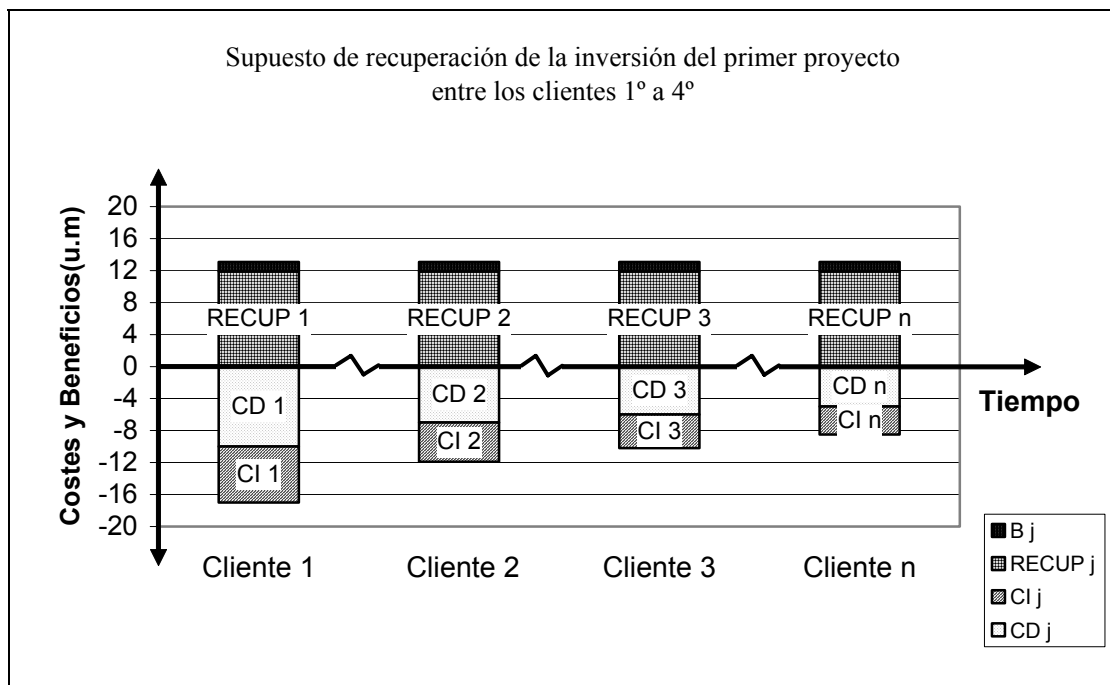


Figura 5. Supuesto de recuperación de la inversión del primer proyecto entre los clientes 1º a 4º (planteamiento estático). Fuente: Basado en [Torralba, 2003].