

Utilización de QFD en la toma de decisiones para la estructuración de una familia de productos*

Montse Ereño¹, Rebeca Cortazar²

¹Depto. de Informática. Universidad de Mondragón

mereno@eps.mondragon.edu

²Departamento de Ingeniería Software. ESIDE, Universidad de Deusto

cortazar@eside.deusto.es

Resumen .La habilidad de una organización para crear productos exitosos depende en gran medida de cómo obtenga, analice y utilice la información asociada al mismo. Para desarrollar un nuevo producto o servicio, es esencial tener en cuenta *“la voz del consumidor”*. En el mercado actual, donde las empresas están siendo emplazadas a entregar mejores productos, con mayor rapidez y a menor coste, a clientes que están demandando producto *“a medida”*, y donde además el Software es un elemento clave en cada vez más productos, la mayoría de las empresas se han replanteado su forma de trabajo y han adoptado una estrategia denominada **“Mass Customization”**. Esta estrategia hace referencia a la fabricación de productos, utilizando técnicas de fabricación en masa, pero dotando a la línea de producción de ciertos grados de libertad, que permita la variabilidad del producto, para generar toda una *Línea de Productos Software (LSP)*. Este nuevo paradigma de desarrollo incrementa y adelanta en el tiempo la toma de decisiones asociadas al desarrollo. Técnicas como QFD (Quality Function Deployment) ayudan en la toma de estas decisiones.

1. Introducción

Las fases iniciales del proceso de desarrollo de Productos Software Intensivos son básicamente una toma de decisiones continua:

- ¿Qué espera el cliente de este producto?
- ¿Qué funcionalidades ofrece este producto Software intensivo?
- ¿Qué aspectos no funcionales son necesarios?
- ¿Cuál es el diseño óptimo para cubrir las necesidades detectadas?
- ...

* La investigación asociada a este artículo ha sido posible gracias a la financiación por parte de la Diputación Foral de Guipuzkoa del proyecto titulado *“MecaInfor: Adaptación de una metodología de estructuración de productos mecánicos al entorno informático”*.

Si en lugar de un único producto, nos enfrentamos a la definición y desarrollo de toda una Línea de Productos Software, las decisiones se multiplican:

- ¿Qué funcionalidades ofrece esta Línea de Productos Software?
- ¿Cuáles de estas funcionalidades son comunes a todos los productos de la línea?
- ¿Cuáles son opcionales?
- ¿Cómo estructuro el producto para soportar de forma robusta esta comunalidad/variabilidad?
- ...

En este artículo presentamos la aplicación del método QFD como herramienta de soporte a la toma de decisiones en el desarrollo de una línea de productos Software Intensivo. El resto del artículo se estructura de la siguiente forma. En la sección 2 se explica el origen y motivación de las Líneas De Producto Software (LPS). En la sección 3 se analiza el proceso de toma de decisiones en este nuevo paradigma y se analiza la aplicabilidad del método QFD. En la sección 4 se presenta el método desarrollado en el proyecto MECAINFOR para soportar la toma de decisiones utilizando QFD. Finalmente, la sección 5 resume las conclusiones obtenidas y presenta las líneas futuras.

2. Las Líneas de Producto Software: Un nuevo paradigma de desarrollo Software

En la Ingeniería de Software, la aparición de un nuevo paradigma no sucede de forma arbitraria. Esto surge de necesidades concretas que permiten generar una evolución en la forma de codificar, diseñar y modelar. Esto es lo que ha sucedido en el caso de las Líneas de Producto Software.

El mercado actual, tan competitivo y cambiante, ha redefinido la forma de trabajar que tienen las compañías. Los clientes podrían considerarse como un gran mercado homogéneo, pero esto no es cierto ya que son individuos cuyos deseos y necesidades individuales, pueden y deben ser determinadas y colmadas para cada uno. Las empresas están siendo emplazadas a entregar mejores productos, con mayor rapidez y a menor coste, a clientes que están demandando cada vez más en un mercado que se caracteriza por palabras como “comercialización en masa” y rápida innovación. El término **“Comercialización en Masa”** hace referencia a la fabricación de productos, utilizando técnicas de fabricación en masa, pero dotando a la línea de producción de ciertos grados de libertad, que permita la variabilidad del producto, para generar toda una gama o familia de productos.

Se hace referencia al mercado como el “Mercado del Comprador” en el que las empresas deben satisfacer los requerimientos para cada cliente. El mercado dominado por el vendedor de los años cincuenta y sesenta estaba caracterizado por una alta demanda y una relativa escasez de suministros. Las empresas produjeron grandes vo-

lúmenes de productos idénticos, basándose en técnicas de producción de masa. El mercado dominado por el comprador desde los años ochenta, esta forzando a las empresas, que fabrican sus productos específicos en grandes lotes, a producir un creciente número de productos ajustados a las necesidades de cada cliente individual, a costa de los beneficios de la fabricación en masa. Por ello, las empresas que ofertan un producto personalizado con un mínimo coste extra, tienen una ventaja competitiva sobre aquéllas que no lo hacen.

Ante esta nuevo escenario, los distintos sectores se han visto obligados a reaccionar buscando “soluciones” que traten de abordar esta problemática. Si nos centramos en el sector del desarrollo Software, vemos que esa “solución” se basa en la **Reutilización**. Esta no es una solución novedosa. Echando la vista atrás, vemos que la historia de la programación es como una espiral creciente donde el elemento clave que origina cada vuelta de espiral o cada salto es la **Reutilización** (Fig. 1). Ya en los últimos años de la década de los sesenta, la idea de construir sistemas mediante la composición de *Componentes Software* fue presentada como solución a la afamada *Crisis del Software* por M. D. McIlroy [1]. Durante la década de los setenta se propugnó la reutilización de los *Módulos*, mientras que en los años ochenta la influencia del *Paradigma Orientado a Objetos* hizo que la *Clase* se convirtiera en la unidad de reutilización. Sin embargo, todas estas tendencias fallaban en conseguir un enfoque sistemático de reutilización. Todas ellas daban lugar a iniciativas individuales, frecuentemente realizadas a pequeña escala. Para solucionar este problema surgen los *Frameworks* [2] y la *Programación Orientada a Componentes* [3].

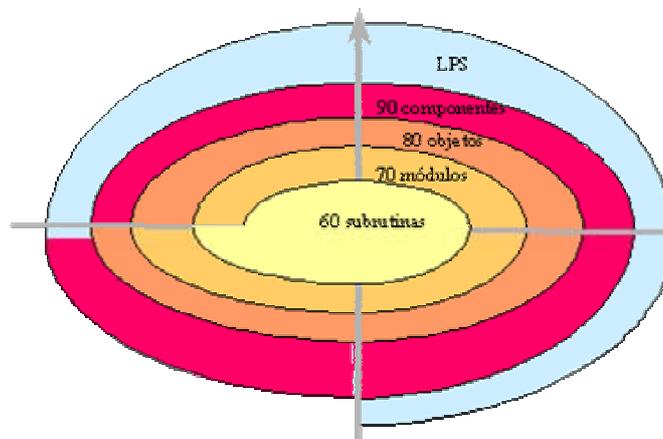


Fig. 1. Evolución en las formas de desarrollo Software

La **Reutilización** del Software es uno de los objetivos fundamentales dentro de la *Ingeniería del Software*, ya desde sus inicios. Actualmente, cada vez son más los contextos empresariales donde el desarrollo Software es visto como una actividad de re-desarrollo. Estamos hablando de un entorno donde las empresas venden sus productos para un uso masivo y permiten la adaptación de éstos a las necesidades cam-

biantes del mercado. Por ejemplo, sectores como el de telecomunicaciones, aplicaciones para el hogar, sector automovilístico, etc. donde se construyen productos iguales o muy similares una y otra vez (y donde la mayoría de los productos han sido ya desarrollados anteriormente).

Así, las **Líneas de Producto Software (LPS)** representan el siguiente escalón o vuelta de espiral. Las LPS se entienden como una de las formas más prometedoras para llevar a buen término un *Plan de Reutilización*, consiguiéndose así un incremento en la productividad, en la competitividad en el mercado y en la calidad de los Productos Software final. Las LPS prometen convertirse en el paradigma dominante de desarrollo Software de este siglo. La flexibilidad de los productos es el nuevo “himno” en el mercado, y el paradigma de LPS promete sistemas adaptables a las necesidades concretas de cada cliente. La clave del éxito en este nuevo enfoque radica en el hecho de que las características comunes compartidas por todos los productos dentro de la línea, sean explotadas para lograr beneficios económicos. [4][5][6][7][8]

Modularidad: La base de las Líneas de Producto

La filosofía que se esconde detrás del enfoque de Líneas de Producto es la estructuración de dichos productos en componentes. La **modularización** o **estructuración de productos** aparece en la literatura relacionada como un término asociado a la *Mass Customization*. La modularización representa la estructuración de un producto en partes, donde estas partes, que contienen funcionalidad, se pueden combinar entre ellas, dentro de una estructura común, dando lugar a diferentes productos. El potencial de los productos modulares es enorme, solo hay que pensar en que la mayor parte de cada futuro producto será creada a partir del ensamblado de partes ya existentes. Sin embargo, la identificación de estas partes resulta una tarea complicada, sobre todo cuando hablamos de sistemas complejos. [9][10].

La idea de construir Sistemas Software mediante el ensamblado de componentes no es nueva, como ya se ha comentado anteriormente. Sin embargo, si se analizan los casos mencionados, se ve cómo muchos de ellos tratan la estructuración de producto focalizando siempre hacia la derivación de productos o hacia la producción. Ninguno de ellos se plantea la estructuración de productos en base a las demandas del mercado, es decir, no tienen en cuenta las **actividades de predesarrollo**.

Son numerosos los trabajos que afirman que afrontar adecuadamente la fase de predesarrollo de los nuevos productos contribuye directamente al éxito del mismo. Sin embargo, en la práctica, se observa que los recursos de las empresas se invierten en mayor grado en actividades posteriores al predesarrollo. Cooper y Kleinschmidt [11] encontraron que el 54 % del total de gastos se destinaba al lanzamiento, frente a un 39 % destinado al desarrollo del producto y a un 7% para actividades de predesarrollo. [12] advierte que las actividades de iniciación apenas ocupan una tercera parte del tiempo total de desarrollo, recomendando a las empresas un mayor esfuerzo en su realización. [13] afirman que la última etapa, de lanzamiento, es la que absorbe el mayor compromiso monetario y la mayor dedicación de la dirección. También [14]

aprecian serias deficiencias en la ejecución de las actividades de predesarrollo por parte de las empresas analizadas.

Está claro pues, que la habilidad de una organización para crear productos exitosos depende en gran medida de cómo obtenga, analice y utilice la información asociada al mismo. Para desarrollar un nuevo producto o servicio, es esencial tener en cuenta “la voz del consumidor”. Hay que conseguir que el cliente articule, estructure y priorice lo que quiere y necesita de un producto o servicio. Una comunicación directa con los clientes permite aprender de ellos y ajustar los productos a sus necesidades. Esto nos da pistas para considerar que debe existir una relación entre la fase inicial de desarrollo, o Ingeniería de Requisitos y la fase de diseño o estructuración de productos, de tal forma que la estructuración de un producto puede y debe ser una consecuencia directa de las necesidades de mercado.

3. Toma de decisiones en el desarrollo de Líneas de Producto Software

La toma de decisiones es el proceso durante el cual una persona debe escoger entre dos o más alternativas. Con frecuencia se dice que las decisiones son algo así como el motor de los negocios y en efecto, de la adecuada selección de alternativas depende en gran parte el éxito de cualquier organización.

En el desarrollo de Productos Software específico para clientes concretos, el proceso de toma de decisiones se puede situar sobre todo en la fase de diseño. Este es el momento en el cual el diseñador debe decidir la forma en la cual traducir las necesidades de los clientes en requisitos de ingeniería o técnicos. En el enfoque de LPS sin embargo, la toma de decisiones se adelanta en el tiempo. Ya no existe un cliente con unas necesidades concretas sobre un producto. En el enfoque de LPS todo el mercado es cliente potencial de nuestros productos. El objetivo del equipo de desarrollo es diseñar un marco que represente a la LPS completa y a partir de él, personalizar los productos individuales. Este enfoque se caracteriza por dos aspectos principalmente:

- Existen numerosos requisitos comunes a toda la LPS, pero otros son específicos de productos concretos
- En el desarrollo están involucrados diferentes roles (Stakeholders.)

Es difícil imaginar un campo de mayor trascendencia para el ser humano que el de la toma de decisiones, y la *Ingeniería de Requisitos* es la fase del desarrollo de Productos Software donde la componente humana es mayor. Esto hace que el proceso de toma de decisiones resulte crítico.

Para tomar decisiones acertadas, es mejor basarse en la frialdad y objetividad de los datos, más que en intuiciones, deseos y esperanzas. Aunque también es cierto que los datos, plantean varios problemas: el modo de obtenerlos, su fiabilidad, el hecho de darles una interpretación adecuada, etc. Hay que establecer cauces de comunicación

para que esta información pueda ser recogida, analizada, resumida y estructurada. En el área de desarrollo de LPS la mayor parte de esta información se encuentra en el mercado.

Por los motivos mencionados, hemos seleccionado el método QFD (Quality Function Deployment) como soporte a la toma de decisiones [15] [16]. QFD es un método sistemático para el desarrollo de productos basados en las necesidades de los clientes. Es un método que involucra a los clientes en el proceso de desarrollo de un producto desde las fases iniciales. Traduce lo que el cliente quiere en lo que la organización produce; esto le permite a una organización priorizar las necesidades de los clientes, encontrar respuestas innovadoras a esas necesidades, y mejorar los procesos hasta una efectividad máxima. De este modo QFD se convierte en una ayuda a la hora de tomar decisiones.

Hemos elegido QFD por sus beneficios tangibles (reducción en el tiempo de desarrollo, detección anticipada de posibles cambios, incremento de la confianza en el diseño, control de los factores económicos) e intangibles (mejora de la satisfacción de los usuarios, base de conocimiento para futuras planificaciones, supone una documentación en sí misma). La tabla 1 resume los motivos por los cuales hemos considerado que este método es adecuado:

Tabla 1. QFD como técnica de soporte en la toma de decisiones

Características de la Toma de decisiones	Cómo aborda QDF este aspecto
No deben existir conflictos acerca del objetivo final.	QFD permite determinar de forma <u>cuantitativa</u> los objetivos.
El tomador de decisiones tiene que conocer los criterios, las alternativas y las posibles consecuencias de su determinación.	QFD se basa en la correlación entre características (Que/Como/Cuanto). Esto permite conocer como una decisión afecta a dichas características.
Preferencias claras: deberían poderse asignar valores numéricos y establecer un orden de preferencia para todos los criterios y alternativas posibles.	QFD trabaja con datos cuantitativos
No se debe desestimar el futuro.	QFD permite visualizar configuraciones futuras
Diferentes perspectivas ayudan a tomar la decisión	QFD permite el trabajo conjunto de equipos de trabajo multidisciplinares.

4. Aplicación de QFD en el desarrollo de LPS

Los objetivos de QFD son:

- Reconocer la correlación entre los requisitos del cliente y las características del producto
- Identificar las características del producto que afectan a requisitos específicos de cliente
- Reconocer las correlaciones entre las características de ingeniería

Para alcanzar esos objetivos, QFD se basa en la construcción de matrices o tablas de calidad donde se confrontan datos desde diferentes perspectivas (Fig. 2) La primera de estas matrices se llama “House of Quality” (HOQ). No existe un número determinado de matrices a utilizar, depende de cada caso. En cada nueva matriz se desciende en el nivel de abstracción, de tal forma que las matrices iniciales utilizan un lenguaje más funcional, y las últimas entran en detalles más técnicos.

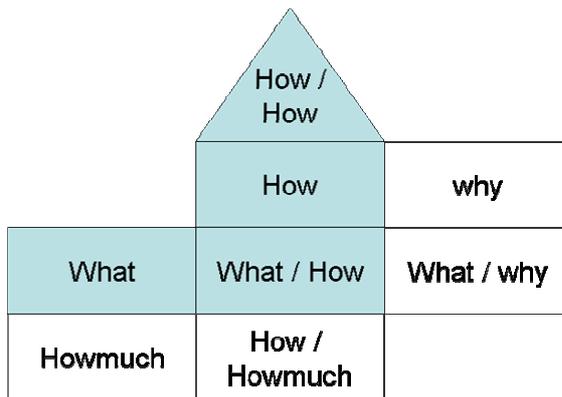


Fig. 2. House of Quality (HOQ)

Tradicionalmente, QFD se venido aplicando en el desarrollo de productos tangibles; es decir, productos físicos. Nuestro objetivo es adaptar esta técnica en el desarrollo de un producto-servicio no tangible. *El Software*. Así, en la aplicación de este método hemos definido unas hipótesis de partida:

- En todo momento pensamos en Productos Software Intensivo. No tenemos en consideración productos donde la componente Software es una mas entre componentes mecánicas, electrónicas, etc.
- Suponemos que las necesidades de mercado, que se utilizan como punto de partida en la aplicación del método, son el resultado de un análisis de mercado profundo y estricto.

Nuestro método se puede resumir en los siguientes tres pasos (Fig. 3):

1º Paso

Como ya hemos dicho, para desarrollar un nuevo producto o servicio, es esencial tener en cuenta “la voz del consumidor”. De este modo el primer paso de nuestro método tiene como objetivo determinar la voz del cliente. Se trata de conseguir que el cliente articule, estructure y priorice lo que quiere y necesita de un producto o servicio. Sin embargo, hay que tener en cuenta, que no existe una única voz de cliente, las voces son diversas y todas ellas deben ser consideradas. En la definición de una línea de productos están involucrados muchos y diversos participantes (usuarios finales con distintas características, desarrolladores con diferentes roles, etc.), a los que genéricamente se les denomina *Stakeholders*. Cada participante o *Stakeholder* tiene su propia perspectiva del futuro producto. Cada una de estas perspectivas, por si sola, resulta incompleta, es una descripción parcial del sistema final. Sintetizar el conocimiento poseído por los múltiples participantes o *Stakeholders* es una de las claves de éxito del desarrollo de nuevos productos. En nuestro método se identifican cuatro participantes o *stakeholders* con sus respectivas perspectivas:

- **Clientes o Mercado:** este grupo representa todos los posibles clientes representativos dentro de un segmento del Mercado.
- **Empresa:** este grupo está formado por todos aquellos roles dentro de una empresa involucrados en el desarrollo y comercialización del producto (Marketing, Diseño, Aseguramiento de la calidad, Producción, Contabilidad, etc...)
- **Diseñadores:** Este rol está relacionado con los aspectos de implementación del producto
- **Stakeholder:** Este rol agrupa a todos los anteriores.

Para obtener todo el conocimiento asociado al producto, los roles anteriores deben ser interpelados. Cada uno de ellos tiene diferentes intereses y los expresa con diferente nivel de abstracción. Para unificar toda esta información, construimos la primera matriz de HOQ donde se Cruzan diferentes vistas (Fig.5). En primer lugar confrontamos las expectativas del rol cliente contra los requisitos del producto desde el punto de vista del rol empresa). En esta matriz se utiliza un nivel de abstracción elevado, el lenguaje es meramente funcional. Tras este paso, se han identificado y estructurado los requisitos de la LPS

2º Paso

El resultado del primer paso es una identificación completa de los requisitos deseados de la LPS. Sin embargo debemos tener en cuenta que existen características o necesidades de un producto que afectan a muchas otras. A estas características se les denomina *Aspectos* o “Crosscutting concerns”. Mas concretamente el término “*crosscutting concerns*” se refiere a los factores de calidad o funcionales del Sistema Software que no pueden ser modularizados de forma efectiva utilizando las técnicas de desarrollo de Software existentes [17] [18] [19]

En los últimos años, los Aspectos se han convertido en un área de creciente interés, y aunque en sus orígenes estaba centrada sobre todo en la fase de programación, en los últimos años ha surgido una corriente, denominada “aspectos tempranos”, que estudia los “crosscutting aspects” desde etapas tempranas, como son los requisitos, y la arquitectura.

En el segundo paso de nuestro método, establecemos la correlación entre la identificación de requisitos obtenida en el paso 1 y los “crosscutting concerns” o *aspectos*. De este paso obtenemos un conocimiento cuantitativo de las características de calidad deseadas por los usuarios, así como los requisitos que se ven afectados y en que grado.

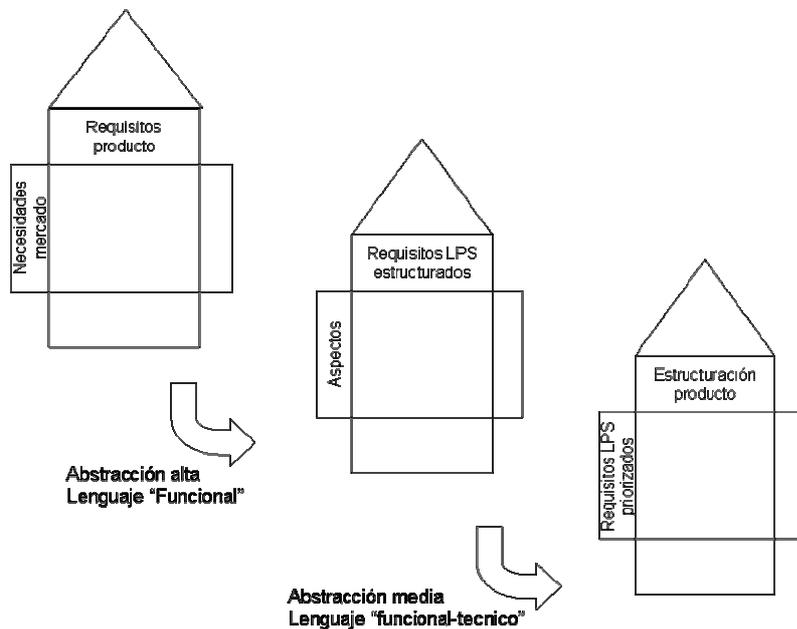


Fig. 3. Método MECAINFOR para la estructuración de LPS

3Paso

En este paso el nivel de abstracción baja. Se comienza a emplear un lenguaje más técnico, dirigido a los diseñadores de la LPS. En este paso se realiza una primera propuesta de componentes. Para realizar esta propuesta, se parte de los conceptos de la programación orientada a objetos: abstracción, encapsulamiento, herencia, polimorfismo, etc. Se establece la correlación entre la propuesta de componentes y los requisitos de producto. Esto nos permite detectar si sobran o faltan componentes. Así mismo, los componentes se correlacionan entre ellos (tejado de la HOQ) basándose en su comportamiento. De esta forma podemos detectar la necesidad de optimizar esta propuesta de componentes, añadiendo o eliminando alguno de ellos.

En resumen, nuestro método ofrece una propuesta de estructuración de producto, basada en técnicas y métodos de Ingeniería Software, y también en datos cuantitativos. El método además proporciona otras ventajas adicionales, como es el hecho de permitir una trazabilidad total de cada módulo con los requisitos origen.

5. Conclusiones y líneas futuras

Este método se ha definido dentro de un proyecto subvencionado parte de la Diputación Foral de Guipuzkoa, titulado “MecaInfor: Adaptación de una metodología de estructuración de productos mecánicos al entorno informático”. Actualmente estamos inmersos en la aplicación de este método en un caso práctico.

Este método está centrado en la estructuración de producto. En un futuro este método se podría ampliar incluyendo las siguientes consideraciones:

- Profundizar en el análisis de mercado inicial, ponderando las diferentes perspectivas.
- Continuar con la aplicación del método hasta la fase de codificación.
- En el enfoque de LPS no sólo se tienen en cuenta las necesidades actuales, sino también las futuras. Es necesario predecir las futuras necesidades del mercado, para ser capaces de integrarlas en los productos de la línea. Resultaría interesante tratar el factor tiempo de forma explícita en el método.

6. Referencias

1. NATO Conference on Software Engineering, (1968) pp. 88-98. "Mass-Produced software components". In J.M. Buxton, P. Naur, and B.Randell, editors, *Software Engineering Concepts and Techniques*.
2. WIRFS-BROCK, R. J. y JOHNSON, R. E. (1990). "Surveying Current Research in Object-Oriented Design". *Communications of the ACM*, 33(9):105- 124.
3. SZYPERSKI, C. (1998). "Component Software – Beyond Object-Oriented Programming". Addison-Wesley.
4. PAUL CLEMENTS, LINDA NORTHROP (2002), "Software Product Lines: Practices and Patterns", Addison Wesley.
5. MARCOS ESTERMAN, KOSUKE ISHII, (2001) "Concurrent Product Development Across the Supply Chain: Development of Integrator/Supplier Risk and Coupling Indices," ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference.
6. LINDA M. NORTHROP (2002)," SEI's Software Product Line Tenets". *IEEE Software*, 19(4):32–40.
7. SOFTWARE ENGINEERING INSTITUTE. (2001). "A Framework for Software Product Line – Version 3.0". Product Line Systems Program, Software engineering Institute. Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 (USA). <http://www.sei.cmu.edu/plp/framework.html>.
8. SVAHNBERG M, BENGTSSON P.(2000), "Software Product Lines from Customer to Code",. Research Report 2000:1, ISSN: 1103-1581, Department of Software Engineering and Computer Science, University of Karlskrona/Ronneby, Sweden,
9. KOSUKE (KOS) ISHII Modularity: A Key Concept in Product Life-cycle Engineering Defining Modules, Modularity and Modularization Thomas D. Miller, Per Elgård
10. D. L. PARNAS," Decomposing Systems into Modules"
11. COOPER, R. G. y KLEINSCHMIDT, E. J. (1988): «Resource Allocation in the New Product Process», *Industrial Marketing Management*, 17, pp. 249-262.
12. PAGE, A. (1998) "Assessing New Product Development Practices and Performance: Establishing Crucial Norms", *Journal of Product Innovation Management*, 10, 4, pp. 273-90.
13. URBAN, G. y HAUSER, J. (1993): "Design and Marketing of New Products", Prentice-Hall, 2ª ed., New Jersey.
14. KHURANA, A. y ROSENTHAL, S. R. (1997): "Integrating the Fuzzy Front End of New Product Development", *Sloan Management Review*, 38, 2, pp. 103-120.
15. HUNT, R. B (1997): "Applying QFD principles to strategic transformation". In: *Transactions from the Ninth Symposium on Quality Function Deployment*. Novi, Michigan, pp. 235-263
16. JAYARAM, J., HANDFIELD, R., AND S. GHOSH (1997), "The Application of Quality Tools in Achieving Quality Attributes and Strategies," *Quality Management Journal*, 5(1): 75-100.
17. C. DUAN, J. CLELAND-HUANG, R. SETTIMI, X. ZOU (2004) "Decision Making support for early aspects identification based on probabilistic trace retrieval "
18. ELENA NAVARRO, PILAR, RAQUELG., ANTONIA REINA, JUANMA,ANA, LIDIA, (2004) " Early aspects" in DSOA2004
19. ELISA BANIASSAD AND SIOBH'AN CLARKE (2004) "Finding Aspects In Requirements with Theme/Doc" EarlyAspect2004