

## TRANSER: Ingeniería de Requisitos en un Desarrollo Orientado al Mercado.

Montse Ereño Incera  
Universidad de Mondragón  
[mereno@eps.mondragon.edu](mailto:mereno@eps.mondragon.edu)

Jokin Arizmendi, Eusebi Calonge  
(Adur Software Productions S.Coop)  
{j.arizmendi,ecalonge}@adur.com

Joxeja Oiarbide  
A2  
[Jjoiarbide@a2.coop](mailto:Jjoiarbide@a2.coop)

### Abstract

*La industrialización del software se configura, desde hace ya algún tiempo, como una de las nuevas vías de desarrollo en el sector de las Tecnologías de la Información. Sin embargo, el mercado todavía está dando sus primeros pasos. La brecha entre el mundo académico y el mundo industrial es considerable dentro de este contexto. Mientras la comunidad investigadora avanza en el estudio de este nuevo enfoque de desarrollo, y en la definición de técnicas y modelos que lo soportan, la aplicación industrial encuentra innumerables escollos en su aplicación. Una de las mayores dificultades se encuentra en las fases iniciales del desarrollo. Es comúnmente aceptado que el éxito de los productos software se mide en base al grado de cumplimiento de las expectativas de los usuarios; por lo tanto, resulta crucial capturar dichas expectativas lo más fielmente posible. Esto, no resulta nada sencillo cuando el producto no va dirigido a un cliente concreto, sino a todo un sector del mercado.*

*Este artículo muestra las conclusiones obtenidas tras la aplicación de una serie de prácticas de Ingeniería de Requisitos Orientada a Perspectivas (IROP) en un caso real de desarrollo orientado al mercado.*

### 1. Introducción

El éxito de los productos software se mide en base al grado de cumplimiento de las expectativas de los usuarios. Desafortunadamente, resulta difícil encontrar a un usuario completamente satisfecho. Esto se debe a que generalmente los usuarios de una aplicación tienen diferentes expectativas de la misma. Si además, nos centramos en el desarrollo de productos software orientados a mercado, la cuestión se complica. Bajo este enfoque, ya no existe un cliente con unas

necesidades concretas sobre un producto, ahora todo el mercado es cliente potencial de los productos. Integrar todas las expectativas sobre el producto se convierte en un área de investigación crítica.

La disciplina, dentro de la Ingeniería Software, encargada de lidiar con los deseos y necesidades de los stakeholders, es la Ingeniería de Requisitos (IR) [1][2]. Tradicionalmente, la IR se ha entendido como una parte borrosa del ciclo de vida software, en la que se obtiene una especificación formal de unas ideas informales. Sin embargo, desde mediados de los años setenta esta fase ha cobrado una especial importancia, y actualmente es considerada la etapa clave en el desarrollo software.

En la Universidad de Mondragón, y más concretamente en el grupo HAZI se lleva tiempo investigando y trabajando en el enfoque de desarrollo orientado al mercado, y más concretamente en las fases iniciales de IR [3][4][5][6]. En el año 2005 comenzó un proyecto de colaboración (TRANSER), subvencionado por el programa GAITEK, que permitió analizar la aplicabilidad de un modelo ideal en un caso real. Este modelo ideal propone la utilización de un enfoque típico del desarrollo a medida, como es la Ingeniería de Requisitos Orientada a Perspectivas (IROP), en un contexto de desarrollo orientado al mercado.

En este artículo se detalla la forma en la cual se enfocó la fase inicial del desarrollo bajo el enfoque IROP, se describen las actitudes generadas y las dificultades encontradas y finalmente se presentan las conclusiones obtenidas desde una perspectiva de investigación. En el apartado 2 se explica el nuevo enfoque de desarrollo software orientado al mercado: las Líneas de Producto Software, y se presenta la problemática de la IR en este enfoque. Así mismo se muestra la relevancia del tema para la comunidad investigadora. En el apartado 3 se explica el enfoque

IROP mientras que en el apartado 4 se desarrolla su aplicabilidad al enfoque LPS desde un punto de vista teórico. En el apartado 5 se muestra el enfoque de investigación adoptado. Los apartados 6 y 7 tratan sobre el desarrollo del proyecto y los apartados 8 y 9 muestran las conclusiones obtenidas tanto desde un punto de vista de investigación como desde una perspectiva de proyecto

## 2. Contexto de la investigación

Con el fin de adaptarse a las nuevas exigencias del mercado, las empresas han cambiado la estrategia de desarrollo software. El nuevo enfoque de desarrollo se denomina: Líneas de Producto Software (LPS) y promete convertirse en el paradigma dominante de desarrollo software de este siglo. 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.

Informalmente una LPS está formada por un conjunto de productos estrechamente relacionados (por su funcionalidad), que son vendidos a los mismos grupos de compradores, que son comercializados a través del mismo tipo de distribución, o que caen en el mismo rango de precios. Una definición formal de LPS es la dada por el SEI[7] “Conjunto de sistemas software que comparten un conjunto de características común y gestionado, que satisface las necesidades específicas de un segmento de mercado y que son desarrollados a partir de un conjunto de activos de una forma establecida”.

Como se dice en la introducción, la disciplina, dentro de la Ingeniería Software, encargada de lidiar con los deseos y necesidades de los stakeholders, es la IR. Bajo el enfoque de LPS, la IR no se centra en los requisitos de un único sistema, sino en los requisitos de toda una familia. Esto supone una serie de cambios importantes. Analizando diversos estudios que comparan ambos enfoques [8] [9] [10] [11] [12] [13] [14] [15] [16], se observa que la mayoría de las diferencias están relacionadas con la participación de los stakeholders en el proceso de desarrollo. En el enfoque de LPS, la implicación y tratamiento de los stakeholders se complica bastante. El problema básico es la comunicación. La comunicación empresa desarrolladora – cliente/mercado, no resulta sencilla debido a que el cliente final de una LPS no es una única persona/entidad, sino que es todo el mercado. Esto presenta un escenario donde

- Los consumidores tienen necesidades múltiples y contrapuestas.

- Las necesidades de los consumidores varían (segmentos)
- Las necesidades se miden mediante escalas no monetarias
- Es complicado evaluar los beneficios (necesidad de predecir la credibilidad)

Este escenario resulta de interés para la comunidad investigadora, y así lo demuestra el hecho de que el tema de la conferencia internacional en IR del 2006 fuera “Entendiendo los deseos y necesidades de los stakeholders”, y el número de Marzo/abril del 2007 de la revista IEEE software se titulara “Stakeholders en la Ingeniería de requisitos”. Este escenario de investigación combina dos temas de interés para la comunidad investigadora en la actualidad: las LPS y la IR. Si nos centramos en las LPS podemos afirmar que son un campo activo de investigación dentro de la Ingeniería Software. Esto se refleja tanto en las crecientes publicaciones sobre el tema [17][18] así como las conferencias y Workshops dedicadas exclusivamente al área, tales como Software Product Line Conference (SPLC), Component-Based Software Engineering (CBSE), Internacional Conference on Component Deployment (CD), Internacional Conference on Software Reuse (ICSR), etc. En cuanto a la IR, su interés se encuentra plenamente justificado por ser ésta la etapa en la cual se encuentran arraigados los problemas más graves que pueden surgir en los proyectos de desarrollo software. Además, se puede afirmar que la investigación en IR se ha consolidado como un área de investigación en desarrollo, y esto lo confirman la organización de los congresos específicos como El IEEE International Symposium on Requirements Engineering (RE) y la IEEE International Conference on Requirements Engineering (ICRE), actualmente unificadas, el Workshop en IR (WER), la publicación de revistas especializadas como el Requirements Engineering Journal, la aparición bianual de números monográficos sobre IR en IEEE Software, la financiación pública de proyectos europeos como: NATURE (Novel Approaches to Theories Underlying Requirements Engineering), REAIMS (Requirements Engineering Adaptation and Improvement for Safety and dependability), CREWS (Cooperative Requirements Engineering With Scenarios), etc.

## 3. Solución propuesta: El enfoque IROP.

EL grupo HAZI lleva tiempo trabajando en el enfoque de IR para LPS [3] [4] [5] [6]. Este trabajo se basa en la posible aplicación de un enfoque de IR ya

existente como es el enfoque IROP para la integración y tratamiento de stakeholders en LPS.

La Ingeniería de Requisitos Orientada a Perspectivas (IROP) reconoce que toda la información de un sistema no puede ser descubierta considerando a dicho sistema desde una única perspectiva. Al contrario debe recogerse y organizarse la información desde puntos de vista diferentes.

El origen de este enfoque es un paso más dentro de la tendencia creciente y natural, en la historia de la Ingeniería Software, a la participación del usuario en la fase de IR. Si nos situamos en los primeros tiempos del desarrollo software, cuando prácticamente todos los desarrollos se realizaban “a medida” y con el claro objetivo de automatizar ciertas funciones, no existía una clara participación del usuario/cliente en el proceso de desarrollo. Ni siquiera se diferenciaba entre usuario y cliente. En aquel momento la preocupación estaba relacionada con demostrar si la participación era buena o no, e incluso se trataba la fórmula de participación. Más adelante, y viendo los problemas de no involucrar al usuario, se detecta la necesidad de la participación del mismo [19] [20] [21]. Según va evolucionando el mercado y la tecnología, la participación de otros roles, además del de cliente y usuario, se va haciendo esencial. Con el objetivo de conseguir un producto de calidad se introducen en el proceso los roles de revisores, inspectores y testadores; con el objetivo de soportar la evolución y mantenimiento de los productos, surge el rol de mantenimiento; aparecen, o al menos se distinguen, los roles de analista, diseñador, arquitecto, jefe de producto, jefe de proceso, etc. Todos estos roles trabajan conjuntamente en el desarrollo del producto. Además, se ve la necesidad de involucrar también en el desarrollo a todas aquellas personas que de alguna forma se ven afectadas por el resultado del mismo: clientes, usuarios finales, etc. Para poder manejar todos estos roles participantes surge el concepto de Stakeholder. Así se define stakeholder como una persona u organización que tiene influencia en los requisitos de un sistema o que es impactado por el sistema [19]. Esto incluye no solo a los usuarios resultantes de la creación de un sistema, sino también a aquellos que originaron el proyecto, aquellos cuyo trabajo o actividades cambiaron por la creación del sistema, aquellos que lo construyeron y aquellos que pueden influir en el éxito del proyecto.

IROP considera a diferentes stakeholders como el origen de la información relacionada con el problema, donde cada uno/a de estos agentes tiene diferentes puntos de vista (todas ellas igualmente válidas pero incompletas) del problema. El enfoque de IROP es el primero en proponer una integración explícita no solo

de los stakeholders, sino también de cada punto de vista de cada stakeholders. Así surge el concepto de perspectiva. A la combinación de stakeholder más su punto de vista u opinión del sistema se le denomina perspectiva.

### 3.1 Estado del arte

Son varios los métodos definidos bajo el enfoque IROP. El primer método donde las perspectivas empezaron a utilizarse de forma implícita es en SADT [22] y aparecieron por primera vez explícitamente en el método CORE [23]. A partir de ese momento son varios los enfoques y propuestas de perspectivas que han ido surgiendo: VOSE [24], PREVIEW [25], VORD [26], LEITE [27]. Todos estos métodos se encuadran en la fase de IR, pero cada uno de ellos hace un énfasis especial en diferentes tareas; así mismo cada método tiene su propia noción del concepto perspectiva. Sin embargo, todos ellos comparten una serie de características:

- Reconocen de forma explícita la diversidad de fuentes origen de requisitos
- Proporcionan una forma de identificar las fuentes orígenes de requisitos o stakeholder así como de chequear sus contribuciones a los requisitos.
- Proporcionan un mecanismo para organizar y estructurar toda esta diversidad de información.

El enfoque de perspectivas es un enfoque relevante. Esto lo demuestra el echo de que no solo se ha utilizado mucho en las fases iniciales del desarrollo software, sino que son varias y diversas sus áreas de aplicación. Por ejemplo se han aplicado en sistemas de enseñanza inteligentes, donde las perspectivas encapsulan conocimiento considerado útil [28]; En el entorno de las comunicaciones, el modelo de referencia ODP define 5 perspectivas (empresa, información, computación, ingeniería y tecnología) a partir de los cuales se especifica el sistema [29]; en el área del trabajo cooperativo soportado por ordenadores (CSCW) las perspectivas se han utilizado para estructurar el análisis a nivel organizacional[30] y en la ingeniería concurrente las perspectivas se han utilizado como un enfoque para gestionar los conflictos [31]. También se ha aplicado este enfoque en la fase de validación del desarrollo software [27]

Actualmente los principios en los cuales se basa el enfoque de perspectivas continúan siendo relevantes en el contexto del desarrollo software. La comunidad del desarrollo software orientado a aspectos (AOSD), aunque ha trabajado mucho la fase de codificación, en los últimos tiempos se están preocupando por los aspectos en las etapas tempranas del desarrollo. Así

mismo, la notación de modelado UML, está añadiendo mecanismos de extensión y permitiendo la modelización de perfiles. Esto permite la elección de la notación más adecuada para cada situación o perspectiva.

#### 4. Actividades clave del enfoque IROP y su alineamiento con las LPS

Se pueden identificar tres tareas básicas y bastante complejas, que todo método de IROP debería soportar:

- **La identificación de perspectivas:** es necesario identificar todas las perspectivas sobre el producto a desarrollar, para así poder obtener el conocimiento completo del mismo. Esta es una tarea esencial y nada fácil. El ingeniero de requisitos debe realizar un balance entre la mayor cobertura ofrecida por un número mayor de perspectivas contra las dificultades de gestión de la información y contra el alto costo del análisis.
- **Tratamiento de la información generada por las perspectivas:** Se debe definir algún mecanismo para soportar la educación y tratamiento de la información generada por las perspectivas
- **Tratamiento de inconsistencias:** Se deben resolver las inconsistencias entre las diversas perspectivas para el desarrollo del producto final. Son varios los autores que proponen la utilización de perspectivas. Estas técnicas permiten representar distintos fragmentos de una especificación y la relación entre dichos fragmentos. La principal ventaja es que se tolera cierto grado de inconsistencia y se retrasa la toma de decisiones.

#### 5. Metodología de investigación

El proyecto TRANSER presenta la posibilidad de aplicar en un contexto real los modelos ideales trabajados en HAZI. No se debe olvidar, sin embargo, que el proyecto TRANSER tiene como objetivo un desarrollo de producto completo, y que en el consorcio formado para lograr este objetivo el rol jugado por la Universidad de Mondragón en este proyecto es el de asesorar en las fases iniciales de desarrollo.

Se ha aplicado un método de investigación cualitativo, basado en la participación, observación y descripción, donde el objetivo ha sido intentar integrar actividades del enfoque IROP en un contexto de desarrollo real orientado al mercado y analizar las actitudes e interpretaciones que el equipo de desarrollo tenía de las mismas

#### 6. Caso práctico

El proyecto sobre el cual se ha realizado el caso de estudio ha sido un proyecto subvencionado por el programa GAITEK durante los años 2005-2007. El proyecto denominado TRANSER tenía como objetivo el desarrollo de una herramienta informática de gestión integral del transporte dirigida a los proveedores de servicios logísticos.

Este proyecto ha sido desarrollado por dos empresas tecnológicas *Adur Software Productions S.Coop.* y *A2*, con el apoyo de la *Universidad de Mondragón (Grupo HAZI)*. El consorcio formado entre tres entidades, dos empresas y un centro de investigación universitario, ha proporcionado un equilibrio adecuado a la realización del proyecto. Las empresas tecnológicas han proporcionado el conocimiento de la actividad de Transporte, basado en 20 años de recorrido, más de 3.000 programas desarrollados en el área, y un mantenimiento actual de más de 100 instalaciones con cerca de 3.000 usuarios. La Universidad de Mondragón, agente de la RVCTI, ha aportado su conocimiento en constante renovación para dar al proyecto el enfoque adecuado, soportado por las tendencias más asentadas en el análisis y desarrollo de soluciones de software, de cara a obtener un producto basado en un análisis detallado de los requerimientos del cliente. Además, ha proporcionado la seguridad necesaria en el cumplimiento del proyecto respecto a las directrices de metodología de desarrollo del momento.

Esta situación ha permitido a la Universidad de Mondragón sumergirse en un entorno real y al mismo tiempo realizar un estudio observacional analizando los comportamientos en cada etapa. Podría así decirse que se ha aplicado un método de investigación descriptivo donde el objetivo ha sido identificar aspectos relevantes de la realidad, concretamente, se han investigado aquellos aspectos relacionados con los conocimientos, las actitudes y las prácticas de Ingeniería de Requisitos y la interpretación que el equipo de desarrollo tenía de las mismas.

##### 6.1 Contexto del proyecto

La nueva herramienta a construir trataba de mejorar el enfoque a la gestión del negocio de las aplicaciones existentes en el mercado en ese momento.

Adur tiene una gran experiencia en implantaciones de software en el cliente final, y ha pretendido definir este producto con una novedad importante. Se pretende desarrollar una solución parametrizable por diferentes distribuidores, y para ello se requiere una gran flexibilidad y simplicidad de instalación.

Las aplicaciones informáticas de gestión del transporte existentes en el mercado están basadas en una concepción enfocada al medio de transporte. Se componen de módulos (marítimo, terrestre, aéreo,...) que gestionan verticalmente todo el proceso de transporte de mercancía a través de ese medio de transporte (Fig. 1) La nueva herramienta priorizará la compatibilidad y la integración entre plataformas y aplicaciones externas, fijándose como objetivo último la eliminación total de los costos de fricción en los flujos de información de la cadena de suministro, independientemente de las aplicaciones informáticas que se usen en cada fase o proceso de gestión.

El análisis de la solución exigía un conocimiento adecuado de las necesidades de los clientes potenciales. Para ello se dio una gran relevancia a las fases previas al desarrollo del software propiamente dicho, centrándose en un amplio Análisis de Requisitos, que fuera capaz de proporcionar un análisis de situación correcto, sobre el que elaborar un nuevo modelo de aplicación flexible y adaptable.

Aunque el proyecto ha consistido en el desarrollo completo de un producto, en este artículo se tratará únicamente la fase inicial de análisis por ser esta la fase donde se ha analizado la aplicabilidad del enfoque IROP

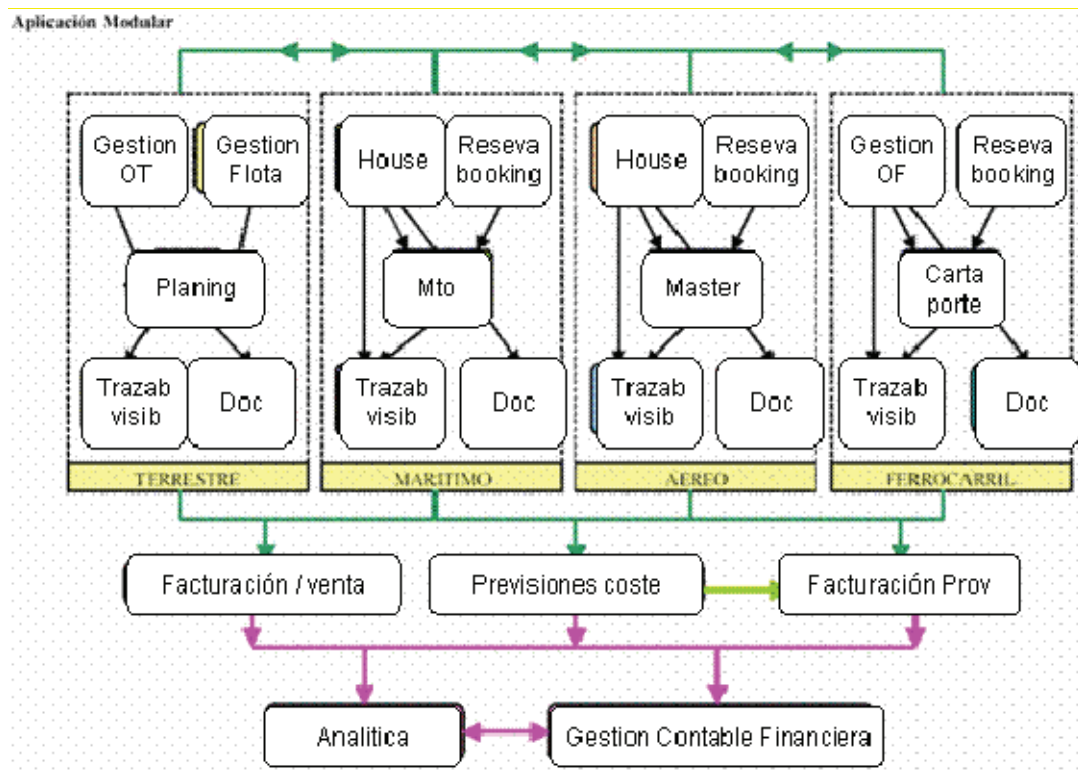


Fig. 1 : Aplicación modular de Gestión del Transporte

## 7 TRANSER Fase 1: Análisis del sistema

El objetivo de esta fase era conocer lo que el cliente quería para ser capaces de producirlo. En este contexto ADUR tiene una gran experiencia en productos implantados, sin embargo, el enfoque del nuevo producto era diferente. Se pretendía desarrollar una solución parametrizable para cualquier actor involucrado en la cadena de suministro. Así se identificaron tres tareas principales dentro de esta fase:

- Tarea 1.1: Identificación de perspectivas.

- Tarea 1.2: Elicitación de perspectivas.
- Tarea 1.3: Análisis y especificación de requisitos.

### 7.1 Identificación de perspectivas

El objetivo de esta tarea fue la identificación de las diferentes perspectivas existentes sobre el producto para asegurar una especificación de requisitos completa y correcta. Esta es una actividad esencial y nada fácil. El ingeniero de requisitos debe realizar un

balance entre la mayor cobertura ofrecida por un número mayor de perspectivas, las dificultades de gestión de la información y el alto costo del análisis.

Teniendo en cuenta que la herramienta en desarrollo pretendía ser un producto homogéneo orientado a un mercado caracterizado por una demanda heterogénea, la actitud debía ser la de centrarse en un grupo específico de consumidores para poder generar más altos márgenes de beneficio. Así en esta tarea se utilizaron técnicas de Segmentación de Mercados que nos permitieron enfocar adecuadamente el mercado potencial de la aplicación y por tanto nos facilitaron la identificación de los clientes potenciales de la misma y sus perspectivas.

La técnica de segmentación toma como punto de partida el reconocimiento que el mercado es heterogéneo y pretende dividirlo en grupos o segmentos homogéneos, que pueden ser elegidos como mercados-meta de la empresa. Así pues, la segmentación implica un proceso de diferenciación de un mercado global en función de múltiples variables y factores.

La finalidad de la segmentación es lograr identificar y conocer el comportamiento real de los consumidores y sus preferencias para satisfacerlas.

Como resultado de esta tarea se desarrolló la matriz de segmentación (ver Tabla 1), estableciendo por un lado los actores que intervienen en la cadena de suministro y por otro las operaciones a realizar para la correcta gestión de toda la información. En dicha matriz se establecen con claridad las áreas de actividad que quedarán cubiertas por el desarrollo de este proyecto, y aquellas áreas que no tendrán cobertura en el mismo, por el momento. La matriz presenta con claridad aquellas áreas de actividad que no se corresponden con los actores de ninguna manera.

La representación de esta matriz nos facilitó la correcta identificación de las áreas en que podía ser adecuado el nuevo producto, así como una definición más clara y adecuada del producto a desarrollar. Además se identificó claramente los stakeholders a los cuales se dirigía el producto y sus características. La propia empresa reconoció nuevos stakeholders que hasta el momento no se consideraban. Lo cual amplió el grupo de perspectivas a tener en cuenta así como sus funciones generales.

Dificultades: la dificultad principal radicó en el miedo de dejar fuera a clientes o perder oportunidad de mercado. Existía una tendencia a querer abordar un

mercado demasiado grande, lo cual se traducían en un producto sin un objetivo claro

## 7.2 Elicitación de perspectivas

En esta tarea el objetivo era la obtención de la voz del cliente (VOC). Esta información conformaría la base de conocimiento para el desarrollo del producto. La técnica utilizada fue la de cuestionarios. Se optó por la utilización de los cuestionarios ya que facilitan el estudio de las necesidades, valores, creencias y motivos de los encuestados, y además permiten estandarizar los datos para un análisis posterior, obteniendo gran cantidad de datos a un precio bajo y en un periodo de tiempo corto

En el transcurso de esta tarea se detectó la necesidad de desarrollar dos cuestionarios diferentes, enfocados a perfiles de clientes distintos.

- Por un lado, resultaba muy interesante conocer las necesidades de nuevos clientes o clientes potenciales para los cual se elaboró un cuestionario corto. Este cuestionario estaba compuesto de 29 preguntas cerradas muy generales sobre las Aplicaciones de Gestión del Transporte y espacios abiertos a las reflexiones sobre las características que debería aportar una nueva aplicación en este campo, así como el nivel de satisfacción con las Tecnologías de la Información utilizadas y las necesidades futuras previstas. El tiempo estimado de elaboración no superaba los 10 minutos. Este cuestionario se rellenó en diversas ferias y eventos del sector.
- Por otro lado, y aprovechando la estrecha relación con clientes ya conocidos, se elaboró un cuestionario más detallado a fin de conocer no solo sus necesidades, sino también su visión de futuro de la actividad. Este cuestionario constaba de más de 100 cuestiones, sobre temas de la actividad y el tiempo de la entrevista se estimó en más de una hora.

La técnica de cuestionarios se alzó como una técnica muy válida para la obtención de información, y se puede concluir que permitió obtener y validar el conocimiento necesario para el desarrollo del producto.

Dificultades: El cuestionario largo, era realmente laborioso de completar. Se utilizó en una feria del sector y se recogieron un reducido número de ejemplares donde además la aportación de información

fue inferior en calidad a la esperada. Por estos motivos el origen de la información finalmente se obtuvo de dos fuentes: por un lado los cuestionarios cortos y por otro lado el conocimiento del propio equipo desarrollador. Para esto último se analizaron los proyectos existentes seleccionando aquellos proyectos que resultaban interesantes bien por la problemática planteada, bien por la solución desarrollada, o por ambas cosas; con esta información se dio respuesta al cuestionario largo.

### 7.3 Análisis y especificación de requisitos

Durante esta tarea, se realizó la descripción del producto. Como resultado de esta tarea se obtuvo el documento de Especificación de Requisitos Software (ERS).

Debido al carácter crucial del documento a originar todo el personal participante en la misma recibió un curso impartido por la Universidad de Mondragón. Dicho curso mostró de forma resumida el contenido y objetivo de las primeras etapas del desarrollo software (la Ingeniería de Requisitos) así como de los resultados o documentos de trabajo que se obtienen y las técnicas y normas existentes en este ámbito. Este curso, aunque bastante general, estaba adaptado al proyecto TRANSER y en ningún momento se consideró como un Estado del Arte de la Ingeniería de Requisitos. El objetivo del curso era unificar los conocimientos del equipo ante una tarea tan importante como esta.

Como resultado del curso se definió la estructura del ERS para el producto TRANSER y las técnicas de modelado a utilizar.

Para definir la estructura del documento ERS más adecuada al producto, se analizaron y compararon estándares de desarrollo software para la especificación de requisitos, como por ejemplo: DoD-DI-MCCR-80025A y IEEE/ANSI 830-1998. A partir de dicho análisis se desarrolló una estructura para el ERS del producto TRANSER. Dicha estructura, pretendió cumplir en todo momento los siguientes principios:

- Separar la funcionalidad de la implementación.
- Crear un modelo intuitivo en lugar de un diseño o modelo de implementación.
- Reconocer que “la especificación debe ser tolerante a un posible crecimiento si no es completa”.
- Desarrollar un modelo del comportamiento deseado de un sistema que comprenda datos y las respuestas funcionales de un sistema a varios estímulos del entorno.

- Establecer el contexto en que opera el software especificando la manera en que otros componentes del sistema interactúan con él.

Para identificar las técnicas de modelado a utilizar, se realizó una sesión de trabajo donde se compararon las dos grandes corrientes de modelado: el análisis estructurado (metodologías funcionales), que es el método de modelado clásico y el modelado orientado a objetos. La elección de los diagramas de modelado a utilizar no fue sencilla. Cada rol participante tenía sus preferencias; por ejemplo, el jefe de proyecto y los analistas se decantaban por lo casos de uso, mientras que los programadores no lo veían interesante y elegían modelos más cercanos al código. Se consiguió una unanimidad total en dos aspectos: la utilización del modelo Entidad-Relación y el compromiso de no elegir demasiados tipos de diagramas. Los criterios que se siguieron a la hora de seleccionar la(s) técnica(s) de modelado fueron:

- Elegir aquellas que reduzcan la ambigüedad.
- Elegir las que resulten fáciles de entender para los stakeholders sin conocimientos informáticos.
- Elegir aquellas que faciliten la validación.
- Elegir aquellas que sean fáciles de modificar.

Finalmente se eligieron los siguientes diagramas como técnica de modelado: Los diagramas de Casos de Uso y el modelo Entidad-Relación para la etapa de Análisis y los diagramas de clases y los diagramas de secuencia para las etapas de diseño y código. La motivación para la elección de dichos diagramas fue principalmente su capacidad de representación y su facilidad de uso.

Dificultades: La mayor dificultad fue la integración de toda la información disponible. Los roles encargados de la especificación fueron un Jefe de Proyecto, un analista y un desarrollador. Por cuestiones de planificación no se utilizó ninguna técnica para el tratamiento de inconsistencias, ni para la priorización de requisitos. Se aprovechó la experiencia del Jefe de Proyecto y del analista para realizar esta tarea. Sin embargo se decidió que el programador fuera una persona sin experiencia previa en el dominio del problema para evitar caer en errores del pasado y ser capaces de proponer una solución nueva.

(a) Actores	Operaciones de gestión			Movimientos de mercancía			Alm	Valor añadido	Gestión Inf.		
	GP	FC	GCI/A	TLDU	TLDM	DC	A	EE	TV	D	
Proveedor materia prima	X	X	X	X	X	X				X	X
Transitorio	X	X	X	X	X	X				X	X
Operador logístico	X	X	X	X	X	X	X	X		X	X
Operador Intermodal	X	X		X	X	X				X	X
Agente de aduanas	X	X	X							X	X
Consolidador marítimo	X	X	X	X	X	X				X	X
Consignatario mercancía LR	X	X		X	X					X	X
Consignatario Buques	X	X		X	X					X	X
Agencias de transporte	X	X	X	X	X	X				X	X
Almacenistas distribuidores	X	X					X	X		X	X
Terminales portuarias	X	X					X	X		X	X
Flotistas (terrestre)	X	X	X	X	X	X				X	X
Courrier o paquetería	X	X	X	X	X	X				X	X
Compañías Aéreas											
Navieras											
RENFE, ... Fabricante											

Tabla 1. Matriz de segmentación

## 8 Resultados del estudio

El objetivo de este trabajo era observar en un contexto real, aspectos particulares a las fases iniciales de un desarrollo software orientado al mercado, y determinar la posibilidad de aplicar prácticas del enfoque IROP.

Una vez finalizado todo el proyecto, no sólo la fase inicial, se han obtenido las siguientes conclusiones:

En cuanto a la identificación de perspectivas se valora positivamente la utilización de técnicas de segmentación de mercado, con participación en el ejercicio de todo el equipo de desarrollo. Se puede concluir que resulta una técnica sencilla y a la vez potente que permite la participación de diversos roles simultáneamente.



En cuanto al tratamiento de la información generada por las perspectivas se han obtenido las siguientes conclusiones

- Se hizo un reparto equitativo de stakeholders. Contrariamente a las fuentes literarias se tuvo en cuenta las perspectivas de los clientes para conseguir un producto rico funcionalmente, y también se tuvieron en cuenta las perspectivas de los desarrolladores, que se preocupaban más de proporcionar una solución técnica innovadora desde las fases iniciales.
- El conocimiento sobre las necesidades de los usuarios era amplio y esto supuso un riesgo, ya que existía una tendencia a “ponerse en el lugar “del cliente, en lugar de comunicarse con él. Esto podría llevarnos a crear un producto tecnológicamente brillante pero no adaptado a las necesidades reales del mercado.
- Se utilizaron los cuestionarios como método de recogida y tratamiento de información. En cuanto a la recogida de información resultaron ser una técnica útil; en este caso la dificultad surgió como consecuencia del aspecto social de la tarea.: resulta difícil conseguir que las perspectivas dediquen tiempo a la elicitación. En cuanto al tratamiento de la información, los cuestionarios no resultaron adecuados. El volumen de información era mucho y no todo quedaba reflejado en los cuestionarios.

En cuanto al tratamiento de inconsistencias se puede afirmar que no se trataron de forma explícita. Esto fue debido a dos cosas: Por una parte el hecho de no tener un método de soporte de información más adecuado que los formularios. Por otro lado, el equipo de desarrollo se sentía en posesión del conocimiento necesario y suficiente para el desarrollo del producto, y realizó un tratamiento de inconsistencias no explícito ni procedimentado

## 9. Conclusiones generales

Una vez finalizado el proyecto, se puede afirmar con rotundidad que es cierto que la Ingeniería de Requisitos es una fase compleja, y en un desarrollo orientado a mercado aún más. Esta fase se ha alargado respecto al tiempo planificado, sin embargo el tiempo dedicado a ella ha sido realmente beneficioso de cara a la definición del producto. Desde el punto de vista de las empresas implicadas en el proyecto, estos beneficios han tardado en visualizarse. Desde la perspectiva de empresa, existía una premura por comenzar la fase de desarrollo. Sin embargo a lo largo del transcurso del proyecto, se ha visto que la

información obtenida y las perspectivas identificadas han ayudado en el desarrollo del resto de las fases.

Por otro lado, y aunque inicialmente el objetivo del proyecto era únicamente el desarrollo del producto, a medida que se realizaba el análisis del dominio, se ha ido viendo la necesidad de definir un proceso de desarrollo y un proceso de gestión de la configuración. Esta necesidad la han planteado las propias empresas implicadas que de forma paralela al proyecto comenzaron su desarrollo

Desde el punto de vista académico, de la Universidad de Mondragón, se ha comprobado que aunque el enfoque IROP resulta viable para realizar un análisis del dominio, resulta pesado de aplicar. La identificación de perspectivas resulta bastante sencilla, sin embargo el manejo de la información asociada a cada perspectiva y el tratamiento de inconsistencias entre perspectivas, necesita de algún soporte metodológico o tecnológico.

En cuanto al equipo de proyecto se puede decir que ha sido una fórmula exitosa. Las empresas disponían del conocimiento del dominio y la universidad del conocimiento del área. Ambas partes han conseguido lograr un equilibrio durante el desarrollo del producto

## 10. References

- [1] Davis, A.M., The art of requirements triage. *IEEE Computer*, 2003, pp: 42-49.
- [2] Damian, D., Stakeholders in Global Requirements Engineering: Lessons Learned from Practice. *IEEE software*, 2007.
- [3] Ereño, M., Landa, U., Cortazar, D. R. Software product lines structuring based upon market demands. In Proceedings of the 2005 Conference on Specification and Verification of Component-Based Systems (Lisbon, Portugal, September 05 - 06, 2005). SAVCBS '05. ACM Press, New York, NY.
- [4] Ereño, M., Cortazar, D. R. Utilización de QFD en la toma de decisiones para la estructuración de una familia de productos. Taller de apoyo a la decisión en Ing. Sw. CEDI Granada, 2005.
- [5] Ereño, M., Cortazar, D. R. Utilización de QFD en la toma de decisiones para la estructuración de una familia de productos. AEMES. *Revista Procesos y Metricas*. Nº5. 2005.
- [6] M. Ereño, R. Cortazar. Eficacia del método ELVIRA-relato de un experimento. XII Jornadas de Ingeniería del Software y Bases de Datos (JISBD) 2007. Zaragoza, 11 al 14 Septiembre 2007 pp.349-354 ISBN: 978-84-9732-595-0
- [7] Software Engineering Institute. A Framework for Software Product Line – Version 3.0. Product Line Systems

- Program, Software engineering Institute. Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 (USA). 2001  
<http://www.sei.cmu.edu/plp/framework.html>
- [8] Bowman, H., Derrick, J., Linington, P. and Steen, M. Cross-viewpoint consistency in Open Distributed Processing. *BCS/IEE Software Eng. J.*, 1996. pp: 44-57.
- [9] Rlshamre, P. A Usability Perspective on Requirements Engineering – From Methodology to Product Development (Dissertation No. 726). Linköping: Linköping University, Linköping Studies in Science and Technology. 2002.
- [10] Kamsties, E., Hörmann, K. and Schlich, M., Requirements Engineering in Small and Medium Enterprises. *Requirements Engineering*, 3, 1998. pp: 84–90.
- [11] Keil, M. and Carmel, E. Customer-Developer Links in Software Development, *Communications of the ACM*, 38, 5, 1995, pp: 33-44.
- [12] Lubars, M., Potts, C. and Richter, C., “A review of the state of the practice in requirements modelling”. In: Proceedings of IEEE Symposium on Requirements Engineering (RE’93), *IEEE Computer Society Press*. 1993.
- [13] Novorita, R.J. and Grube, G. “Benefits of structured requirements methods for market-based enterprises”. In Proceedings of Sixth Annual International INCOSE Symposium. Seattle, WA: INCOSE. 1996.
- [14] Potts, C., “Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software”. In Proceedings of the Second IEEE International Symposium on Requirements Engineering. Los Alamitos, CA: *IEEE Computer Society Press*. 1995. pp. 128–130
- [15] Yeh, A., “Requirements Engineering Support Technique (REQUEST) – A Market Driven Requirements Management Process”. In Proceedings of the Second Symposium on Assessment of Quality Software Development Tools, Los Alamitos, CA: *IEEE Computer Society Press*. 1992. pp: 211–223
- [16] Robertson, S. and Robertson, J., *Mastering the Requirements Process*. Harlow, England: Addison-Wesley. 1999.
- [17] Clements, P. and Northrop, L., *Software Product Lines: Practices and Patterns*, Addison Wesley, 2002
- [18] G. Bockle, P. Clements, J. D. McGregor, D. Muthig, and K. Schmid. Calculating ROI for Software Product Lines. *IEEE Software*, 21(3), May/June 2004. pp: 23-31.
- [19] Gallivan, M.J. & Keil, M. “The User-Developer Communication Process: A Critical Case Study”, *Information Systems Journal*, Vol. 13, Issue 1. 2003. pp. 37-68
- [20] Keil, M. & Carmel, E., “Customer-developer links in software development”, *Communications of the ACM*, Vol. 38, Issue 5. 1995. pp. 33-42.
- [21] S. Kujala, “User Involvement: A Review of the Benefits and Challenges”, *Behaviour & Information Technology*, Vol.22, No. 1. 2003. pp.1-16, *Behaviour & Information Technology*, Vol. 22, No. 1, pp.1-16
- [22] Ross, D. and Schoman, K.E., Structured Analysis for Requirements Definition, *IEEE Trans. Software Engineering*. 3 (1), 1977. pp. 6-15
- [23] Mullery, G.P., “CORE-a method for controlled requirement specification”. En Proc. 4th Int. Conf. on Software Engineering. *IEEE Computer Society Press*. 1979. Pp: 126-135.
- [24] Easterbrook, S. and Nuseibeh, B., “Using viewpoints for inconsistency management”, *Software Engineering Journal*, 11(1), BCS/IEE Press, January 1996. pp: 31-43.
- [25] Sommerville, I., *Software Engineering*, Addison-Wesley, 1992
- [26] Kotonya, G. and Sommerville, I. Requirements Engineering with viewpoints. *Software Engineering Journal*. 1996. 11(1). Pp: 5-11.
- [27] Leite, J.C.P. and Freeman, P.A, Requirements validation through viewpoint resolution. *Transactions of Software Engineering Notes*, 12 (12), 1991. pp: 1253-1269
- [28] Moyse, R. “VIPER: The design and implementation of multiple viewpoints for tutoring systems”. In *Knowledge Negotiation* (R. Moyse and M. T. Elsom-Cook ed.). 1992. Academic Press, London
- [29] Bowman, H., Derrick, J., Linington, P. and Steen, M. Cross-viewpoint consistency in Open Distributed Processing. *BCS/IEE Software Eng. J.* 11(1). 1996. pp: 44-57
- [30] Hughes, J., O’Brien, J., Rodden, T., Rouncefield, M. and Sommerville, I. “Presenting Ethnography in the Requirements Process”. In Proc. Proc. RE’95, York. 1995. pp: 27-35.
- [31] Klein, M. Conflict Management in Concurrent Engineering (special issue). *Concurrent Eng. J.*, 2(3). 1992