



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Departament de Sistemes Informàtics i Computació  
Universitat Politècnica de València

## Método para la especificación de valor en procesos de negocio y la derivación incremental de servicios cloud

Trabajo Fin de Máster

Máster Universitario en Ingeniería y Tecnología de Sistemas Software.

**Autor:** Javier Jiménez Gómez

**Tutores:** Emilio Insfran Pelozo y Fernando González Ladrón de Guevara

2015-2016

# Resumen

---

En la actualidad, los tiempos de entrega de productos software se han reducido drásticamente, siendo lo más habitual entregas parciales sucesivas del software antes que una única entrega final. Uno de los factores que ha favorecido esta tendencia son las nuevas tecnologías del cloud que facilitan el desarrollo de software iterativo e incremental. Facilitando el despliegue de servicios, en cada iteración se despliegan nuevos servicios y funcionalidad de manera incremental. Este hecho ha contribuido a que los tiempos de salida al mercado se reduzcan, optando por el despliegue primero de la parte nuclear del producto software y añadiendo el resto de funcionalidad en las consiguientes iteraciones. Es por ello cada vez más importante que cada iteración del producto contenga los elementos (funcionales o no funcionales) que entreguen el máximo valor al cliente.

Sin embargo, ¿a qué nos referimos con “valor”? En este trabajo se propone un método que permita la especificación del concepto de valor asociado a los procesos de negocio de la organización de manera que dicho valor sirva para la priorización de tareas de negocio en el contexto del desarrollo incremental de servicios cloud dirigido por modelos.

En este trabajo se combinan los conceptos del desarrollo del software ágil y la ingeniería de modelos para proponer un método que permita identificar los elementos de valor que el cliente y el resto de stakeholders creen relevantes para el producto software. El software debe alinearse con los objetivos del negocio para entregar el máximo valor al cliente, es por ello que esta definición de valor se utilizará en las fases iniciales de la ingeniería de requisitos para identificar los requisitos funcionales y no funcionales relevantes y en las fases siguientes del desarrollo como elemento para la priorización de requisitos software. Todo ello integrado en un proceso de desarrollo ágil.

En este trabajo se propone un método que permita guiar la representación del valor y sus relaciones con los modelos de proceso de negocio y de requisitos funcionales. A continuación se establece un criterio para la cuantificación del valor que permita la priorización de las tareas de negocio. También se definen asociaciones entre los distintos modelos (procesos de negocio y valor) y por último se realiza una comprobación inicial de la validez de la definición del valor y el método propuesto por medio de una prueba de concepto.

**Palabras clave:** valor, desarrollo dirigido por modelos, desarrollo ágil, procesos de negocio, priorización

# Abstract

---

Nowadays software delivery timing has been reduced drastically. Continuous software delivery is no more common than a unique final delivery. The new cloud tendency have stimulated the use of iterative and incremental software development. As the cloud facilitates the service deployment, each software iteration incrementally deploys new services and functionality. This fact has contributed to the reduction of the time to market. Now is common to opt for initially deploying the core functionality and then incrementally adding the remaining functionality. Consequently, choosing which components are to be developed first, because they contributed more to the client's value, is a critical decision.

However, what do we understand by "value"? This work proposes a method for the value specification associated to the organization's business processes, so that value serves as a prioritization element of the business tasks in an incremental, model driven cloud services' development.

The concepts of agile development and requirements engineering are combined in this project to propose a method that allows the identification of value elements, relevant to the stakeholders of product software. Software should align with business' value objectives so that the final product delivers the most value to the client. That is the reason why the value specification will be used in the initial phases of the requirements engineering. The value specification should allow to identify relevant functional and non-functional requirements and to work as an element of the software requirements prioritization. All this method, integrated within a agile development process.

A guide to the representation of value and its relationships with business and requirement models is proposed in this work. A criteria to the value quantification for the prioritization of the business tasks is presented. Mappings between value and business models are also proposed. Finally the validity of the method proposed is initially assessed using a proof of concept.

**Keywords:** value, model driven development, agile development, business processes, prioritization.

# Agradecimientos

---

Quisiera agradecer en este trabajo a mis tutores Emilio y Fernando por la ayuda en la elaboración de este trabajo sin la cual no podría haber sido posible. A su vez agradecer a todos los integrantes del proyecto Value@Cloud por su ayuda en uno u otro momento y por crear un buen clima de trabajo. Por último, no por ello menos importante, a mi familia, amigos, compañeros de piso y de laboratorio, por convivir conmigo en las fases más difíciles de este trabajo.

## **Índice de contenidos**

Índice de contenidos.....	5
Índice de tablas.....	7
Índice de figuras .....	8
1. Introducción.....	10
1.1 Planteamiento del problema.....	11
1.2 Objetivo del trabajo .....	12
1.3 Trabajos relacionados .....	13
1.4 Contexto .....	16
2. Marco conceptual.....	18
2.1 Desarrollo de software ágil.....	18
2.2 Procesos de negocio .....	21
2.3 El concepto de valor.....	24
2.3.1 Perspectiva de negocio.....	25
2.3.2 Perspectiva de la ingeniería del software.....	27
2.4 Alineamiento entre IT y negocio: El valor como elemento cohesionador.....	29
2.5 El Modelado de valor .....	30
2.6 La priorización de requisitos dirigida por el valor.....	39
2.6.1 Estimación de importancia.....	40
2.6.2 Estimación de esfuerzo .....	42
3. Propuesta .....	46
3.1 El modelado de valor.....	46
3.1.1 Objetivo .....	46
3.1.2 Selección de un lenguaje de representación del valor .....	47
3.1.3 GRL en profundidad .....	49
3.2 Vista general del proceso de desarrollo dirigido por el valor .....	50
3.2.1 Iniciación .....	52
3.2.2 Especificación del valor.....	53
3.2.3 Priorización de Objetivos .....	54
3.2.4 Diseño del proceso de negocio de alto nivel de abstracción .....	55
3.2.5 Estimación de esfuerzo.....	56
3.2.6 Priorización de actividades del proceso de negocio .....	57
3.2.7 Selección de actividades candidatas para la iteración .....	58
3.2.8 Refinamiento de procesos de negocio candidatos .....	59
3.2.9 Reestimación de esfuerzo.....	60
3.2.10 Reselección de actividades candidatas para la iteración.....	61

3.2.11	Descomposición de actividades en casos de uso.....	62
3.2.12	Selección de casos de uso para la iteración (preparación del sprint).....	63
3.2.13	Desarrollo de la iteración.....	64
3.2.14	Análisis de la iteración.....	65
3.3	El proceso de modelado del valor .....	65
3.3.1	Definición de actores .....	66
3.3.2	Modelado de los actores .....	67
3.3.3	Modelado del actor sistema .....	68
3.3.4	Discusión y validación.....	68
3.4	Implementación de los métodos de priorización .....	68
3.4.1	Priorización de tareas .....	69
3.5	Uniando procesos de negocio con la especificación del valor.....	73
3.5.1	Selección de un lenguaje de representación de los procesos de negocio.....	73
3.5.2	La unión de modelos de valor y de procesos de negocio en la literatura .....	75
3.5.3	Una guía para la transformación del modelo de objetivos en modelo de proceso de negocio.....	78
4.	Caso de estudio .....	80
4.1	Definición: Agencia de Viajes.....	80
4.2	Aplicación del método de modelado del valor.....	81
4.2.1	Iniciación .....	81
4.2.2	Especificación del valor.....	82
4.2.3	Priorización de objetivos.....	94
4.2.4	Diseño del proceso de negocio a nivel alto de abstracción .....	96
4.2.5	Estimación de esfuerzo .....	100
4.2.6	Priorización de actividades del proceso de negocio.....	101
4.2.7	Selección de actividades candidatas para la iteración .....	101
5.	Conclusiones.....	103
5.1	Resultados .....	103
5.2	Valoración del trabajo.....	104
5.3	Trabajo futuro .....	104
	Referencias .....	106

## **Índice de tablas**

Tabla 1 Relaciones entre el actor sistema y el actor agencia de viajes .....	93
Tabla 2 Relación entre elementos del actor sistema y el actor cliente .....	93
Tabla 3 Relaciones entre agente intermediario y Entidad financiera y sistema .....	94
Tabla 4 Importancias calculadas .....	95
Tabla 5 Elementos intencionales del sistema ordenados por valor .....	95
Tabla 6 Asociaciones entre los elementos intencionales de los actores de GRL con las actividades de las calles de GRL .....	99
Tabla 7 Actividades y su esfuerzo relativo estimado. ....	101
Tabla 8 Resultado del cálculo de la prioridad sobre las actividades del proceso de negocio. ....	101
Tabla 9 Actividades ordenadas por prioridad. ....	102

## Índice de figuras

Figura 1 Coste relativo de arreglar un error según la fase en la que ha sido encontrado [1]. .....	10
Figura 2 Logo del proyecto Value@Cloud.....	16
Figura 3 Ejemplo de diagrama UCM [49]......	24
Figura 4 Ejemplo anotado de diagrama e <sup>3</sup> -value [10].....	34
Figura 5 Metamodelo simplificado de VDML [59]......	36
Figura 6 Vistas y elementos relacionados en VDML [60]......	37
Figura 7 Vista de colaboración. ....	37
Figura 8 Vista de red de valor.....	38
Figura 9 Vista de red de actividades.....	38
Figura 10 Vista de la jerarquía de colaboración. ....	39
Figura 11 Vista de la participación de roles. ....	39
Figura 12 Estimación aproximada de orden de magnitud a lo largo de un proyecto [65]. .....	44
Figura 13 Diagrama de actividad en el que se enumeran las distintas actividades del método y su orden secuencial.....	51
Figura 14 Iniciación .....	52
Figura 15 Especificación del valor .....	53
Figura 16 Priorización de objetivos. ....	54
Figura 17 Diseño del proceso de negocio de alto nivel de abstracción.....	55
Figura 18 Estimación del esfuerzo.....	56
Figura 19 Priorización de actividades del proceso de negocio. ....	57
Figura 20 Selección de actividades candidatas para la iteración.....	58
Figura 21 Refinamiento de procesos de negocio. ....	59
Figura 22 Reestimación del esfuerzo. ....	60
Figura 23 Reselección de actividades candidatas para la iteración. ....	61
Figura 24 Descomposición de actividades en casos de uso.....	62
Figura 25 Selección de casos de uso para la iteración.....	63
Figura 26 Desarrollo de la iteración.....	64
Figura 27 Análisis de la iteración. ....	65
Figura 28 Vista general del proceso de modelado de valor.....	66
Figura 29 Vista general del proceso de la tesis doctoral de [71]......	76
Figura 30 Modelo de valor e <sup>3</sup> -value.....	81
Figura 31 Representación de los actores en GRL.....	82
Figura 32 Modelado de los elementos intencionales de la agencia de viajes.....	83
Figura 33 Modelado de los elementos intencionales del actor cliente.....	84
Figura 34 Modelado de los elementos intencionales del actor agente intermediario. ..	85
Figura 35 Modelado de los elementos intencionales del actor entidad financiera. ....	85
Figura 36 Modelado de los elementos intencionales del actor compañía de transportes. .....	86
Figura 37 Modelado de los elementos intencionales del actor compañía de alojamiento. .....	86
Figura 38 Modelado de las relaciones del actor agencia de viajes.....	87
Figura 39 Modelado de las relaciones del actor cliente. ....	88
Figura 40 Modelado del agente intermediario y su relación con las compañías de alojamientos y transportes. ....	89
Figura 41 Modelado del actor entidad financiera.....	90



Figura 42 Modelado de los elementos intencionales del actor sistema .....	91
Figura 43 Representación final en GRL. ....	92
Figura 44 Actores modelados como piscinas en BPMN.....	97
Figura 45 Modelado de las actividades del actor cliente.....	98
Figura 46 Modelado de las actividades del actor sistema .....	98
Figura 47 Modelado de las actividades del actor entidad financiera .....	98
Figura 48 Modelado de las actividades del actor agencia de viajes .....	98
Figura 49 Modelado de las actividades del actor agente intermediario .....	98
Figura 50 Vista general del diagrama de proceso de negocio en alto nivel de abstracción. .....	100

## 1. Introducción

Un objetivo de la ingeniería del software es aportar soluciones tecnológicas a los problemas de la sociedad. Para hacerlo, estas soluciones deben primero basarse en particularidades del problema, las necesidades de los implicados y su contexto. Cuando estas soluciones no tienen en cuenta adecuadamente estos factores, la ingeniería del software falla y los productos creados son desechados.

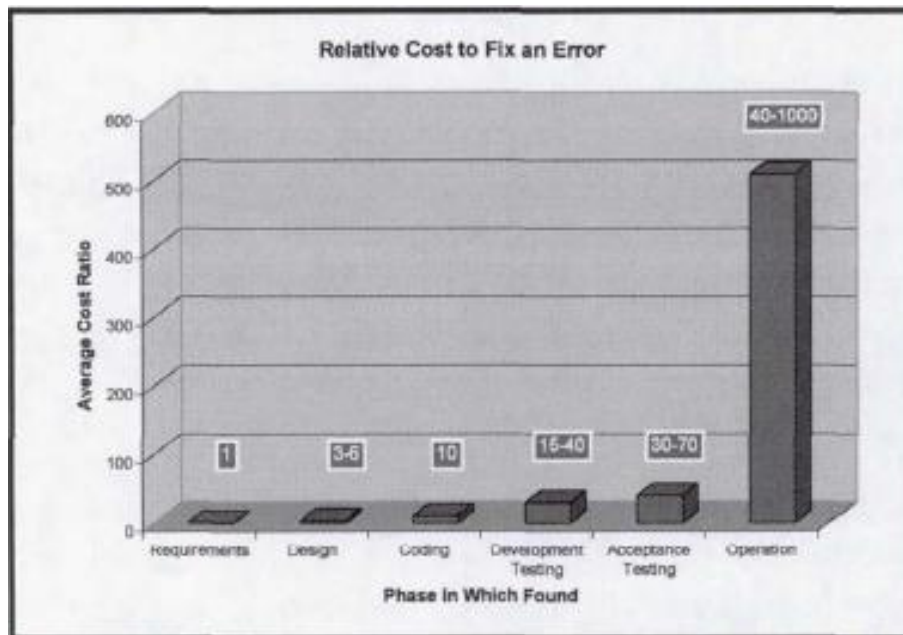


Figura 1 Coste relativo de arreglar un error según la fase en la que ha sido encontrado [1].

Debido a ello, una parte fundamental de la ingeniería del software es la ingeniería de requisitos, que es la encargada del estudio de la extracción, representación y gestión de los requisitos funcionales y no funcionales de un producto software. Como se ha visto en multitud de ocasiones, la fase de elicitación de requisitos es una fase crucial dentro del desarrollo de software ya que de su éxito en buena medida dependerá el éxito del producto software. En la mayoría de desarrollos software los mayores contratiempos se producen por errores no corregidos a tiempo en los requisitos software que llegan incluso a comprometer el éxito del desarrollo, ya que errores en una fase tardía incrementan los costes de desarrollo, alargan los tiempos de entrega y necesita de esfuerzo adicional necesario para corregir las inconformidades (re-trabajo) [1].

Los procesos de desarrollo tradicionales tienen una amplia historia de expectativas no cumplidas y fracasos a la hora de entregar software que no satisfacía o que solo parcialmente cubría las necesidades del cliente [2]. Analizando el problema se reveló que es muy difícil descubrir todas las necesidades del cliente en una fase inicial de elicitación de requisitos ya que en muchas ocasiones incluso el propio cliente no sabe, en el momento inicial, qué necesita, y su concepción del producto cambia cuando ve prototipos o el producto finalizado. De esta manera se popularizaron los desarrollos ágiles e incrementales donde las funcionalidades requeridas del producto software se iban elicitando, diseñando, produciendo y testeando por ciclos; de manera que las

necesidades de cambios y nuevos requisitos fueran más fáciles de incluir en el ciclo de vida del proyecto, reduciendo de esa manera su coste.

A fin de entregar un software que cumpla con los requisitos del cliente, se entregue a tiempo y se desarrolle en un entorno abierto al cambio, diferentes prácticas ágiles se introdujeron hasta formar paquetes metodológicos que englobamos bajo el término de “desarrollo de software ágil” que fueron promovidos por los firmantes del *Agile Manifesto* [4]. En ellos se hace especial hincapié en que el software entregado debe aportar valor al cliente y aportar valor lo antes posible. Pero, ¿a qué se refiere el término “valor”?

El concepto de valor para el cliente todavía está poco explorado en el ámbito de la ingeniería de software aunque si se ha trabajado extensamente en las ciencias económicas. ¿Cómo debe alinearse el valor definido por los modelos de negocio con las soluciones TI? Esta es una pregunta en la que se está trabajando aportando frameworks de trabajo con el valor, lenguajes para su representación y metodologías para emplearse como punto inicial y guía del desarrollo software. Esta es la intencionalidad detrás de la *Value Based Software Engineering* [3].

En la actualidad, el tiempo de entrega se ha reducido considerablemente y la nube y las arquitecturas de servicios permiten que el software se produzca de forma más flexible, rápida y se despliegue de forma incremental, es por ello que conseguir encontrar los requisitos que más valor aporten al cliente en cada iteración del proceso de desarrollo es crucial para mantener la competitividad considerando que el mercado demanda cada vez tiempos de entrega más cortos y donde la diferencia entre un entregable y otro puede marcar el éxito de un producto.

Una de las tendencias en el desarrollo de software que ha pretendido acabar con la informalidad en la representación de los requisitos y que ha tratado de automatizar el proceso de desarrollo de desarrollo de software ha sido la Ingeniería de Modelos. Esta rama de la Ingeniería de Software trata de representar los sistemas software en forma de modelos (representaciones abstractas del conocimiento y/o actividades relativas a un dominio) y utilizar estas representaciones de manera automática o semiautomática para construir productos software; cubriendo con ellas totalmente el proceso de desarrollo.

En el presente trabajo se propondrá un método que tratará de afrontar el proceso de elicitación de requisitos utilizando el valor del cliente e integrando esta información en un proceso de desarrollo que combine elementos de las prácticas ágiles, la ingeniería del software basada en el valor y la ingeniería de modelos con el fin construir software que proporcione valor del cliente.

## **1.1 Planteamiento del problema**

Los modelos de producción basados en la nube y/o en servicios y los actuales tiempos de entrega de los productos software requieren de un método capaz de identificar, de forma ágil y precisa, los requisitos del software a desarrollar. De manera que las sucesivas entregas de software satisfagan las necesidades más inmediatas y de más valor para el cliente.

En primer lugar, es necesario dotar de suficiente expresividad en la captura de requisitos, es decir, es necesario hacer explícitas las necesidades del cliente, qué metas deben

conseguirse con el producto de las Tecnologías de la Información y la Comunicación (TIC) que necesita la organización y cómo éstas contribuyen dentro del proceso de negocio de la organización. Antes de enfocarse en el qué, debe enfocarse en el porqué de los requisitos. Solo de esta manera se conseguirá que el sistema de información se alinee con los procesos de negocio de la empresa y solo así reportará valor para la organización.

También es posible que estos requisitos no hayan sido capturados con precisión o incluso que la idea del producto que el cliente presenta inicialmente evolucione con el tiempo y sea necesario adaptar el producto a estos nuevos requisitos. Este factor de cambio ha sido estudiado extensamente por la literatura que ha tratado de reducir el problema en parte introduciendo métodos de desarrollo iterativos e incrementales, en los cuáles, el producto se desarrolla incrementalmente en partes más pequeñas, y por tanto, son validadas en esos incrementos por el cliente pudiendo aportar realimentación sobre el producto y así alimentar con este conocimiento la siguiente iteración. Sin embargo, a pesar de esto se siguen produciendo errores que acarrear costes de desarrollo, elementos del sistema que no responden a las necesidades de los usuarios y cambios inesperados. Estos errores suelen tener su fuente en los requisitos por lo que una minimización de estos ayudará a disminuir los costes de producción y obtener un producto de más valor para el cliente desde una fase temprana de desarrollo.

El desarrollo iterativo es una constante también en los métodos de desarrollo ágiles que como expresa el *Agile Manifesto* [4], tratan de “entregar software funcional y de valor para el cliente desde una fase temprana”. Para ello, las prácticas de este proceso ágil tienden a involucrar al cliente en el proceso de desarrollo de software, hasta el punto de localizarlo junto al equipo de desarrollo, a fin de obtener información del cliente de manera temprana y así introducir la perspectiva de este en el desarrollo del producto de manera activa.

A pesar de ser un método muy válido, en la práctica ha encontrado dificultades a la hora de involucrar a personal de la organización cliente en el proceso de desarrollo (ya sea porque esta organización no puede prescindir de un recurso humano para que se integre con el equipo de desarrollo o porque no se puede tener acceso directo a un representante de la organización de forma regular) o incluso una vez involucrados, el cliente no es capaz de aportar su conocimiento al proyecto debido a que el nivel de abstracción no es lo suficientemente alto [7]. Un ejemplo concreto ocurre durante el proceso de priorización de requisitos y objetivos, en el que el cliente debe elegir entre un conjunto de requisitos identificados para seleccionar aquellos que aportan más valor y que deben ser implementados de manera preferente. En muchas ocasiones el cliente tiene claro qué influencia tendrá en su negocio el sistema de información pero tiene dificultades en aplicar ese conocimiento a unidades más pequeñas del software.

Los conceptos de valor son vitales para potenciar el éxito del desarrollo del producto software así que es importante disponer de mecanismos para expresar estas necesidades y prioridades de forma explícita y que influya directamente en las actividades de desarrollo. Se debe aprovechar el conocimiento del cliente en aquellas partes del dominio en la que es experto y facilitar su integración en un proceso ágil de desarrollo de software

## **1.2 Objetivo del trabajo**

El principal objetivo de este trabajo es definir un método que conjugue las herramientas, prácticas y técnicas del desarrollo de software ágil, el desarrollo de software dirigido por

modelos y el desarrollo de software dirigido por valor con el propósito de disponer de un método de definición del valor del cliente utilizando modelos que se integren dentro de un proceso de desarrollo ágil, y que finalmente, guíen el desarrollo del software a través de la priorización de aquellos elementos que produzcan mayor valor para el cliente.

Este trabajo pretende primero identificar un lenguaje de modelado con la capacidad de expresión suficiente para la descripción del valor del cliente. Para ello se analizará en la literatura las aproximaciones existentes que soporten la idea de modelado del valor. A su vez se determinará qué enfoque para la definición del valor se utilizará, de manera que esta sirva como lazo conceptual entre las perspectivas del valor de negocio y la ingeniería de software.

Una vez identificado este lenguaje de descripción de valor usando modelos, se propondrá la adaptación del lenguaje y los procesos de modelado asociados a él para adaptarlo a las características y necesidades de este proyecto (trazabilidad entre los diferentes elementos, facilidad para su descomposición en incrementos, etc.).

En segundo lugar, el fin de este modelado del valor es que sea útil para la identificación y definición de requisitos, así como también su priorización según estos contribuyan al valor del cliente. Por ello, el modelado del valor tiene varias perspectivas: *exploratoria*, trata de descubrir que es lo importante para el cliente, qué es lo que este espera del software desarrollado y qué necesidades pretende cubrir con él; de *trazabilidad*, los requisitos software deberán derivarse de estas necesidades de valor del cliente y deberán poder relacionarse con estas; y de *priorización*, el valor debe dirigir la priorización de los requisitos en las fases posteriores del desarrollo de manera que las entregas al cliente aporten desde el primer momento el máximo valor.

En tercer lugar, en este trabajo se estudiará cómo derivar requisitos software partiendo de la base de la definición del valor del cliente, utilizando distintos niveles de abstracción desde los procesos de negocio de alto nivel, la arquitectura software necesaria y finalmente los casos de uso que utilizarán los desarrolladores como punto de partida para la programación.

En último lugar, la propuesta de este proyecto pretende ser pragmática y adaptable a las prácticas habituales utilizadas por los desarrolladores en contexto de metodologías ágiles existentes como Scrum [5], XP[6] y otros.

### **1.3 Trabajos relacionados**

En la literatura se han encontrado diferentes aproximaciones al problema del modelado del valor. Desde diferentes ramas del conocimiento se ha estudiado el problema, siendo la de economía y empresa la que más ha trabajado en el concepto de valor a través de modelos de negocio y diferentes refinamientos como el caso de la Cadena de Valor [37], en estos se profundizará más en la sección 2.3. Mientras que su estudio dentro de la ingeniería del software, algo más tardío, tiene su origen en la profundización en la captura de requisitos y la inclusión de requisitos no funcionales como en el *NFR Framework* [39].

Esta sección se centrará en identificar aquellos trabajos con objetivos similares, es decir, que empleen el valor en las primeras fases de la elicitación de requisitos y/o su uso como elemento de priorización de los requisitos. En la literatura, se observa que no

siempre se ha representado el valor de una manera estructurada en modelos, pero no por ello esas contribuciones son menos relevantes, siendo aplicables con modificaciones a un entorno de desarrollo basado en la ingeniería de modelos.

Se distinguen, *grosso modo*, dos aproximaciones al problema: basadas en lenguajes de definición del valor y basadas en la cuantificación del valor.

Aproximaciones basadas en lenguajes de definición del valor. Una de las primeras aproximaciones al modelado del valor fue la aportada en [9] por Eric Yu (1995) en su tesis doctoral y más tarde (1997) publicada [8]. En ella introduce en el análisis de requisitos la noción del *¿Por qué quiere el cliente este requisito?* señalando los problemas de centrarse solamente en el *qué* y cómo ayuda el *¿por qué?* a entender mejor el dominio. En este trabajo sostienen que este enfoque pretende : ayudar al cliente a entender y definir por sí mismo los requisitos; ayudar al diseñador a contribuir en el rediseño de procesos de negocio, ayudar a prever el cambio dentro del desarrollo, crear información estructurada sobre los dominios y a entender cómo los sistemas cooperan entre sí. Como solución de modelado a la representación de las propiedades y capacidades de razonamiento apropiadas para una fase temprana de la ingeniería de requisitos, se propuso por Eric Yu en estos trabajos, el marco de trabajo *i\** (leído “i estrella”). En él se incluía una ontología y características de soporte al razonamiento diferentes a las de la ingeniería de requisitos en las fases más tardías. En la sección 2.5.1 se presenta una exposición detallada de la versión más reciente del marco de trabajo de *i\**.

Otra aproximación, heredera de la Cadena de Valor y de los modelos de proceso de negocio es la propuesta por Gordijn *et al.* en [9] denominada *e<sup>3</sup>-value*. Esta propuesta, incluye un lenguaje gráfico para la definición del valor el cual trata de representar las proposiciones de valor ofrecidas por los distintos implicados en el sistema de información y las relaciones económicas establecidas entre ellos. Todo ello desarrollado en el contexto de las aplicaciones innovadoras de comercio electrónico.

Este trabajo, propone un lenguaje de modelado que también da soporte a las fases iniciales del proceso de desarrollo: la exploración del problema. En esta representación tratan de representar las diferentes perspectivas del problema (marketing, negocio, tecnologías de la información) en una sola representación. De manera que esta aporte información que permita escoger entre diferentes alternativas, la más económicamente viable. También tratan de aprovechar el poder del modelado para concretar ideas, de manera que esas ideas abstractas de comercio electrónico innovadoras puedan madurarse dentro de una representación estructurada. Igual que en el caso anterior, este diseño en la fase temprana de la elicitación de requisitos sería el punto de partida para la Ingeniería de Requisitos y el resto de fases del proceso de desarrollo. Siendo los modelos de *e<sup>3</sup>-value* los que permitan la elicitación, el análisis y la validación de requisitos. Esta perspectiva híbrido entre la perspectiva de negocio y de ingeniería de requisitos de la ingeniería del software así como el modelo semi-formal que da soporte a su expresión están más desarrollados en la sección 2.5.3 de este trabajo.

La idea de usar el valor como guía del proceso de desarrollo de software ha creado una rama propia en la ingeniería del software, la denominada “*Value Based Software Engineering*” (VBSE, Ingeniería de Software basada en el valor). Biffel *et al.* (2006) [3] recopilan el estado del arte sobre este campo en el libro *Value Based Software Engineering*. En él, defienden que la VBSE lleve las consideraciones de valor a la posición

principal, de manera que las decisiones de la ingeniería del software, a todos los niveles, sean tomadas utilizando estas consideraciones como referencias. Es decir, que las decisiones tomadas en el proceso de desarrollo tengan como fin reflejar los objetivos explícitos de los stakeholders envueltos, tanto del personal de marketing, los analistas de negocio, arquitectos y expertos de calidad, como de los expertos de proceso, directores de proyecto y ejecutivos.

Guiados por estos objetivos, todos los implicados tratan de derivar algún beneficio del desarrollo de software: tangible o intangible, económico o social, monetario o utilitario, o incluso estético o ético. En *Value Based Software Engineering* se establece lo que es valor dentro del proceso de desarrollo de software, extendiendo la definición ISO de ingeniería del software y se crea un marco ontológico sobre el valor. A su vez se recopila el estado del arte de las prácticas que utilizan valor dentro de todos los procesos de la ingeniería de software, desde la valoración de proyectos, cuantificación de riesgos y valor económico, y elicitación de requisitos a la validación del cliente y su contribución a los procesos de negocio.

Una aproximación al uso del valor en el campo del desarrollo de software dirigido por modelos es la realizada por De Castro *et al.* [11][12]. En esta aproximación se trata de resolver el problema de obtener servicios y especificaciones de procesos de negocio relevantes en escenarios de negocio reales. Este problema de alineamiento entre elementos de negocio de alto nivel y Tecnologías de la Información (TI) se trata de resolver utilizando modelos CIM (*Computation Independent Models*, Modelos Independientes de Computación) de manera que se acorte la relación entre el modelado del sistema y del negocio. Para ello se emplea un modelo de valor CIM expresado en e<sup>3</sup>-value. En este trabajo se proponen guías de transformación para convertir ese modelo de valor CIM en modelos de proceso de negocio PIM (*Platform Independent Models*, Modelos independientes de la plataforma) este caso en notación BPMN, que representan el software desde el punto de vista del negocio. Todo ello en el contexto de la arquitectura orientada a servicios (SOA).

En el trabajo de De Castro, se contribuye creando metamodelos tanto para los modelos CIM como PIM adaptados al entorno SOA así como las relaciones de asociación entre ellos. A su vez, se propone el proceso mediante el cual se usarán estos modelos y se integrarán en el proceso de desarrollo. Para ello creando los modelos propuestos y pasando por su desglose como modelos de casos de uso, la creación de los modelos de arquitecturas de servicios derivados, y la generación de los servicios. Todo ello mediante técnicas transformación y derivación de modelos.

En el Trabajo de Fin de Grado (TFG) de Guzmán [13] “Especificación de requisitos para servicios cloud dirigido por valor”, se proponía una primera aproximación para acercar la vista de negocio y la de sistema de información definiendo un método para la especificación de requisitos de servicios cloud incremental y dirigida por valor. Para ello, tomando como partida un modelo de proceso de negocio, se realiza la representación del valor en un diagrama e<sup>3</sup>-value que utiliza en conjunto al modelo de proceso de negocio para derivar los servicios de negocio modelados como casos de uso, que serán implementados posteriormente como servicios en la solución.

En lo que respecta a métodos no estructurados en forma de modelos para la representación del valor, en [14] se propone un método para medir el valor y el coste de

un requisito de manera relativa, utilizando para ello la técnica de comparación por pares *Analytical Hierarchical Process* (AHP). Esta técnica se usa sobre una lista de requisitos en la que se valoran las dimensiones de coste y valor de cada requisito. Después de este proceso se obtiene una lista de requisitos ordenada según el coste y valor relativo que aporten al proyecto.

En la técnica de AHP se valoran los requisitos por pares, el uno con el otro, entre todos los requisitos, valorando cual es menos, 1, o más, 9, relevante/costoso. Aunque esta técnica no profundiza en lo que significa valor para el cliente, la involucración del cliente en el empleo de la técnica y la confianza en su intuición (y la de los técnicos que forman parte del proceso de priorización) hacen que se obtenga un elemento más de información sobre el cual realizar la priorización.

En [15], también con el propósito del uso del valor como elemento de priorización de requisitos software, se establece un marco de trabajo llamado *Value Oriented Prioritization* (VOP). VOP define un mecanismo para cuantificar y ordenar requisitos para un incremento de una aplicación, un prototipo o una especificación de requisitos software. Para ello en VOP se establece una matriz donde los requisitos candidatos se ponderan según contribuyan a las áreas de negocio nucleares (también valoradas), o también según los riesgos que presenten. Finalmente la suma total ponderada de riesgos y su contribución a valores de negocio permite ordenar los requisitos software candidatos.

VOP fue puesta a prueba en una empresa colaboradora donde la aplicación de la técnica dio como resultado un orden de requisitos satisfactorio para todos los *stakeholders*. Esta técnica pretende tener en cuenta los aspectos de valor con un bajo coste, de forma que pueda ser aplicada en empresas de tamaño pequeño y mediano. Técnicas similares a estas se encuentran en el contexto del desarrollo de software ágil donde técnicas como el *planning póker*[68] tratan de establecer un consenso sobre el valor relativo de cada requisito que se introduce en el desarrollo.

#### **1.4 Contexto**

Este Trabajo de Fin de Máster (TFM) está enmarcado dentro del proyecto “Desarrollo Incremental de Servicios Cloud Dirigido por Modelos y Orientado al Valor del Cliente” (Value@Cloud) que es un proyecto de investigación financiado por el Ministerio de Economía y Competitividad en el Programa Estatal de I+D+i Orientada a los Retos de la Sociedad. El proyecto está dirigido por la Dra. Silvia Mara Abrahao Gonzales.



**Figura 2** Logo del proyecto Value@Cloud



Value@Cloud es una aproximación para abordar el reto Economía y Sociedad Digital. En concreto el proyecto se centra en la investigación, desarrollo, innovación y adopción de soluciones y tecnologías ligadas al cloud computing.

Dentro de este proyecto de investigación, este TFM se sitúa en el paquete de trabajo 1, el cual tiene por objetivo: “Definir un marco de gestión para la obtención de valor por medio de la mejora incremental de los procesos críticos de negocio”. En este trabajo se han acometido las tareas de plantear modelos para la ayuda de la planificación estratégica que permitan explicitar los objetivos empresariales; diseñar herramientas y modelos para seleccionar los procesos críticos con mayor impacto en los objetivos de negocio y evaluar lenguajes de dominio específicos que permitan representar procesos críticos de negocio

Durante el desarrollo de este TFM se ha tenido en cuenta el contexto de aplicación tanto por el enfoque iterativo, ágil y dirigido por modelos del conjunto del proyecto como también su integración con los demás paquetes de trabajo definidos en el proyecto de investigación.

Los productos desarrollados en el paquete 1, y llevados a cabo como parte de este TFM, son el punto de entrada para un proceso de desarrollo de servicios cloud incremental, y de reconfiguración dinámica de arquitectura de servicios, cuyos aspectos de calidad y valor entregados al cliente podrán ser monitorizados y utilizados para guiar los posibles cambios en el software o los requisitos si los niveles de calidad, o el valor entregado al cliente, no son los adecuados.

## 2. Marco conceptual

Este capítulo tiene por objetivo exponer los distintos conceptos en cuyo contexto se posiciona este trabajo. En primer lugar se expone los principios del desarrollo de software ágil así como las prácticas y filosofía habituales en la actualidad. A continuación se contempla el concepto de valor desde la perspectiva de negocio y de la ingeniería del software, profundizando en las particularidades de la postura ágil en este campo. A continuación se contempla como la unión entre IT y negocio pasa por comprender los elementos de valor y como esta unión debe resultar de provecho para la organización. Después se exponen qué técnicas de modelado, parte del desarrollo dirigido por modelos, aparecen en la literatura para modelar el valor. Por último, se presenta el estado del arte sobre la priorización de requisitos dirigida por el valor.

### 2.1 Desarrollo de software ágil

Bajo el término de Desarrollo de Software Ágil se recogen un conjunto de principios de la ingeniería de software basados en el desarrollo iterativo e incremental donde los requisitos y soluciones evolucionan mediante la colaboración de grupos auto-organizados y multidisciplinares. Estos principios favorecen la rapidez, la entrega de valor temprana y la flexibilidad como respuesta al cambio. Los procesos de desarrollo ágiles se oponen a los métodos tradicionales siendo la flexibilidad y la autonomía propiedades importantes para los métodos ágiles. De manera que en los desarrollos ágiles, a diferencia de los tradicionales el conjunto del proyecto no es planeado y planificado de antemano, si no que el proceso de desarrollo se divide en varias iteraciones en las cuales se diseñan e implementan partes del sistema software a desarrollar.

El término ágil es utilizado por primera vez dentro de la comunidad de desarrollo de software en 2001 en el llamado *Agile Manifesto* [4] firmado por diecisiete personalidades de la industria y la investigación del campo del desarrollo de software. En él se enunciaban los principales valores y principios de esta forma de desarrollo de software, que marcarían las futuras prácticas que se integrarían en esta disciplina.

Valores ágiles expuestos en el *Agile Manifesto*:

**Individuos e interacciones** sobre procesos y herramientas

**Software funcional** sobre documentación exhaustiva

**Colaboración con el cliente** sobre negociación del contrato

**Respuesta al cambio** sobre seguir un plan

La tendencia general es la de poner por delante los individuos y sus interacciones a los procesos y las herramientas, de manera que estas últimas no se conviertan en el fin si no en un medio para el proceso de desarrollo de software. De la misma manera, se favorece el desarrollo de software que funcione sobre extensas documentaciones que no contribuyan a crear software funcional. Un aspecto muy relevante es la colaboración con el cliente al que se le incluye dentro del proyecto y no solamente como el firmante del contrato y el receptor del producto. De la misma manera, dado que la experiencia

muestra que los cambios se producirán a lo largo del desarrollo se prefiere responder a él sobre seguir una planificación de forma estricta realizada de antemano.

El manifiesto ágil surge después de que a principios de los años noventa un conjunto de métodos de desarrollo “ligeros” comenzaran a surgir como alternativa a los métodos “pesados” anteriormente aparecidos y cuya principal crítica era que eran demasiado rígidos, altamente regulados, rígidos y micro-gestionados que comprometían los resultados del software, poniendo más esfuerzo en el cumplimiento pormenorizado de las tareas del proceso que en el desarrollo del software propiamente dicho. Estos nuevos métodos como Scrum (1995), Cristal Clear (1996), Extreme Programming (XP) (1996) o el diseño dirigido por características (*Feature Driven Development*) (1997) utilizaban conceptos de la gestión de proyectos y desarrollo de software anteriores como el diseño iterativo, la producción evolutiva de software y el desarrollo de software adaptativo. Estos métodos de desarrollo a pesar de ser anteriores al manifiesto se encuentran clasificados como métodos ágiles actualmente por ajustarse a los principios de estos.

12 principios ágiles por el Manifiesto Ágil [17].

1. Nuestra más alta prioridad es satisfacer al cliente a través de **entrega de software de valor temprana y continua**.
2. **Bienvenidos los cambios en los requisitos**, incluso tardíamente en el desarrollo. Los procesos ágiles capturan el cambio para conseguir las ventajas competitivas del cliente.
3. Entregar frecuentemente software funcionando.
4. Gente del negocio y desarrolladores deben trabajar juntos diariamente durante el proyecto.
5. **Construir proyectos en torno a individuos motivados**. Darles el entorno y apoyo que necesiten, y confiar en ellos para conseguir hacer el trabajo.
6. El método más eficiente y efectivo para transmitir información hacia y dentro de un equipo de desarrollo es la **comunicación cara a cara**.
7. Software funcionando es la medida principal de avance.
8. Los procesos ágiles promueven el **desarrollo sostenible**. Los promotores, desarrolladores, y usuarios deberían ser capaces de mantener una paz constante indefinida.
9. La atención continua a la excelencia técnica y el buen diseño mejora la agilidad.
10. **Simplicidad**, el arte de maximizar la cantidad de trabajo NO realizado, es esencial.
11. Las mejores arquitecturas, requisitos, y diseños emergen desde **equipos auto-organizados**.
12. Mejora continua.

Según los principios del manifiesto, y en las metodologías desarrolladas bajo ellos, se puede generalizar que el principal objetivo es la entrega de software de valor, desde el comienzo del proceso de desarrollo y entregándolo de forma continua. Aunque el concepto de valor parezca evidente, en la sección 2.3, se discutirán los problemas de este concepto y las diferentes formas en las que se ha interpretado por distintos autores.

También es característico de las metodologías ágiles su aceptación de los cambios en los requisitos, a diferencia de otras metodologías (clásicas o de cascada) que se centran en que los requisitos para el producto deben de estar fijados y diseñados desde el inicio del

proyecto. Esta propiedad considera que la construcción del software es iterativa y que el descubrimiento de los requisitos debería serlo también.

Una característica fundamental, y que se aborda con más detalle a continuación, es la localización de especialistas del negocio junto a los desarrolladores. De forma que desarrolladores y especialistas del negocio, colaboren estrechamente en la solución TIC se alinee con los objetivos de negocio.

En la práctica, como métodos ágiles se han constituido un conjunto de procesos de desarrollo que utilizan prácticas, roles y herramientas que siguen estos principios ágiles y que se engloban bajo la terminología de “procesos de desarrollo ágiles”. Algunos de los más conocidos son Scrum, Kanban, *Extreme Programming* (XP), etc. Aunque es habitual que en muchas organizaciones el método de desarrollo no se implemente por completo tal y como se especifica, si no que se adapta a la organización, seleccionando entre las prácticas ágiles que resulten más apropiadas o útiles en el contexto de esta [16].

En [16] también se analizan las prácticas más extendidas entre los practicantes de los métodos ágiles. En este trabajo se encuentra que XP y Scrum son las prácticas más utilizadas, y se procede a analizar el grado de implementación de las mismas. En este trabajo, se indica que aunque estas metodologías no se implementan completamente, los participantes encuentran satisfactorios sus efectos en el proceso de desarrollo.

Las prácticas más extendidas entre los profesionales indicadas en [16] son:

1. Desarrollo iterativo.
2. Reunión de planificación del sprint.
3. Refactorización.
4. Reunión diaria de Scrum.
5. Reunión de revisión del sprint.
6. *Product Backlog*.
7. Diseño incremental.
8. Línea de código base única.
9. Iteraciones/*Sprints*.
10. Gráficas *Burndown*.
11. Desarrollo dirigido por pruebas.
12. Integración continua.
13. *Backlog* del sprint.
14. Posesión colectiva del código.
15. Cliente localizado con el equipo.
16. Programación por parejas.
17. Pruebas de aceptación.

Entre estas prácticas se incluyen aquellas relativas a la gestión de proyectos así como otras relativas a la producción del código. El contexto ágil de este Trabajo de Fin de Máster (TFM) viene dado por el proyecto marco bajo el que se engloba, por lo que el trabajo a realizar tiene como requisitos estos principios de agilidad y sigue varias prácticas básicas de la gestión de proyectos.

A continuación se procede a describir con más detalle las prácticas que se contemplan en este TFM y a las que serán referidas de forma recurrente posteriormente:

- **Desarrollo iterativo:** Aproximación al desarrollo basada en un proceso cíclico de prototipado, testing, análisis y refinamiento de un producto software. Tiene como propósito desarrollar un sistema a través de ciclos repetidos de las actividades del proceso de desarrollo. A diferencia del desarrollo tradicional en cascada, en el que se recogen los requisitos por adelantado, se diseña hasta a un gran nivel de detalle, se codifica, se hacen las pruebas y se entrega el producto terminado por completo; el desarrollo iterativo propone varios ciclos en estas actividades, llegando a darse en ocasiones al mismo tiempo [18][19] y se entrega en cada ciclo no el producto completo sino una parte del mismo.
- **Diseño incremental:** Concepto ligado al desarrollo iterativo, en cada ciclo se desarrolla una porción más pequeña del producto y que suele denominarse “incremento”.
- **Iteraciones/Sprint:** Cada uno de los ciclos de desarrollo iterativos e incrementales en los que se desarrolla el software. Cada ciclo incluye las fases básicas del desarrollo de software: Requisitos, Diseño, Implementación, Verificación (Pruebas) y mantenimiento. Estos ciclos se desarrollan en un período reducido de tiempo, habitualmente de una duración de entre una semana y un mes. El término Sprint es propio de Scrum pero es equivalente al clásico iteración.
- **Product Backlog:** Lista priorizada de tareas para el equipo de desarrollo que es derivada de la hoja de ruta y los requisitos. Los ítems de trabajo más importantes se colocan al principio. El equipo de desarrollo no trabaja directamente con esta lista, sino que extrae un conjunto de ítems de trabajo de ella sobre los que trabaja en un sprint/iteración, mientras que los encargados de crear estas tareas, el cliente y el encargado de dirigir la visión del producto, añaden estas al *backlog*, de manera que no entorpezcan el trabajo actual del equipo de desarrollo.
- **Sprint/Iteration Backlog:** Subconjunto de ítems de trabajo asignados al equipo de desarrollo para realizar en un período de tiempo, la duración de un sprint o iteración [20].
- **Cliente localizado con el equipo:** El cliente es una parte vital del proceso de desarrollo por ello se le incluye en él teniéndolo próximo y siendo sujeto activo del proceso de desarrollo, es el encargado de definir las necesidades del producto y validar los resultados.

## 2.2 Procesos de negocio

Un proceso de negocio es una colección de actividades estructuradas o tareas que desempeñadas por un rol sirven a un objetivo particular. La gestión de procesos de negocio (*Business Process Management*, BPM) consiste en el entendimiento y la gestión de diversos procesos organizacionales e inter-organizacionales los cuales conectan a personas y sistemas automatizados. La gestión de procesos de negocio integra métodos, técnicas y herramientas para apoyar el diseño, ejecución, gestión y análisis de procesos de negocio operacionales.

El principal objetivo de la gestión de procesos de negocio no es simplemente la coordinación de cadenas de actividades, si no conseguir que esas actividades trabajen juntas para llevar a cabo un objetivo de negocio. En este trabajo, dichos objetivos tratarán de representarse en un modelo de valor para poder analizar qué procesos o tareas están

contribuyendo (y en qué medida) a los objetivos de negocio, es decir, están creando valor.

El diseño y la gestión de procesos de negocio no son conceptos novedosos pero las tecnologías de la información han permitido facilitar la gestión de los procesos y la automatización de los mismos.

En este trabajo, se abordarán los procesos de negocio como herramientas de diseño de sistemas de información y como fuente para el refinamiento de requisitos y también como forma de comunicación con el cliente.

En la literatura se encuentran diferentes lenguajes de descripción de procesos de negocio y flujos de trabajo. Se pueden dividir, de manera general, en lenguajes centrados en el modelado de dominio y los lenguajes centrados en la ejecución.

Del primer grupo podemos encontrar *Event-driven Process Chain* (EPC) [21], *Business Process Model Notation* (BPMN) [22], *Use Case Maps* (UCM) [23], y *Yet Another Workflow Language* (YAWL) [24] y del segundo grupo, lenguajes centrados en la ejecución, específicamente *Business Process Execution Language* (BPEL) [25], redes de Petri [26] y acercamientos basados en reglas como ECA (*Event Condition Action*, Evento Condición Acción)[27]. La diferencia principal entre ambos grupos se encuentra en el nivel de detalle: los lenguajes de ejecución permiten especificar aspectos más técnicos con mayor detalle y por ello son capaces de ser ejecutados automáticamente. Aunque la frontera entre ambos grupos es cada vez más difusa ya que ambos tipos de lenguajes cuentan con una capacidad de expresión suficiente para representar cualquier tipo de flujo o cuentan con las extensiones necesarias o las transformaciones para convertirse en lenguajes ejecutables.

Un modelo de proceso de negocio presenta 5 perspectivas de modelado. La *perspectiva funcional* se centra en la ejecución de tareas concretas y típicamente considera conceptos para describir subprocesos y tareas atómicas. La *perspectiva de comportamiento* expresa la secuencia de los elementos del modelo. En la *perspectiva organizacional* se destaca quién o qué procesa las tareas y como se gestionará la distribución de las mismas. La *perspectiva informativa* expresa los elementos de datos relacionados con el proceso ya sean necesarios para el procesamiento de tareas o generados por estas. Por último, la *perspectiva de contexto* refleja una visión del conjunto o una meta perspectiva del proceso, y contiene elementos relevantes a las características del proceso como objetivos de negocio o métricas (KPIs).

Los modelados de proceso de negocio se centran en el área funcional, en este TFM se pretende dar a esta expresión del *qué se va hacer*, un contexto de objetivos de negocios y dependencias de usuarios que permita suplir la falta de expresividad de los lenguajes de procesos de negocio en la perspectiva de contexto.

Así pues, para el caso que nos ocupa para la notación de este modelo de procesos de negocio se presentan dos fuertes candidatas: BPMN (estándar y altamente popular y ampliamente utilizado en la especificación de requisitos de desarrollo de software [28] y UCM (también estándar y con una estrecha relación con el modelado de objetivos). En la sección 3.5.1 se razona que lenguaje de representación se empleará en la propuesta y los motivos de su elección.

*Business Process Model and Notation* (BPMN) es un estándar del Object Management Group (OMG), originalmente creado por la *Business Process Initiative*, y desde 2005 mantenida por la OMG. Su última versión, 2.0.2, fue lanzada en enero de 2014 y es actualmente la vigente.

BPMN es una notación para la representación gráfica de flujos de procesos de negocios en un Diagrama de Procesos de Negocios (BPD) para su diseño e implementación. Está basado en diagramas de flujo similares a los diagramas de actividad de UML (*Unified Modeling Language*). Su principal objetivo es soportar la gestión de procesos de negocio tanto por personal técnico como por personal del dominio de negocio. Combinando una semántica intuitiva para los usuarios no técnicos pero lo suficientemente expresiva para representar elementos del proceso de negocio más complejos. También provee una asociación entre elementos gráficos de la notación y una capa subyacente para lenguajes de ejecución como BPEL.

Por su parte, *Use Case Maps* (UCM) forma parte como GRL del estándar de la *International Telecommunications Union* (ITU)[52] denominado URN (*User Requirements Notation*) [23].

La notación de UCM es una aproximación orientada a escenarios que permite describir relaciones causales entre responsabilidades entre uno o más casos de uso. Esta notación permite ilustrar los componentes opcionales envueltos en un escenario en un diagrama similar a un mapa.

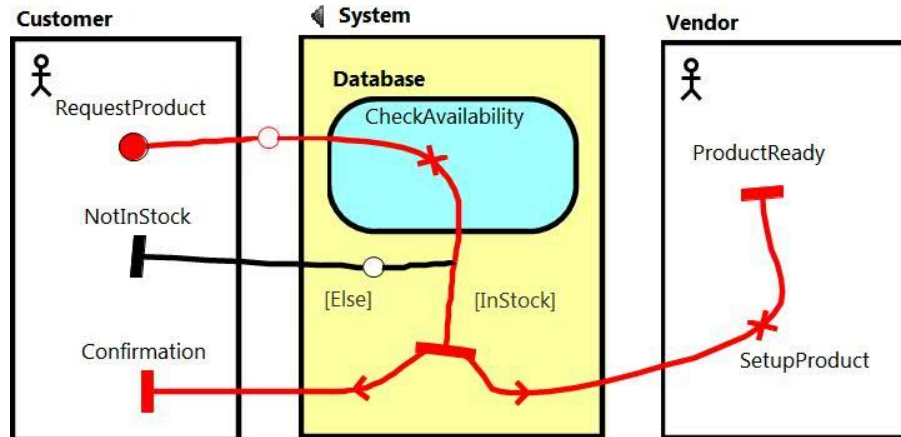
UCM utiliza los recorridos de escenarios para ilustrar la relación causal entre responsabilidades. UCM provee una vista integrada del comportamiento y la estructura permitiendo la superposición de caminos de escenario en una estructura de componentes abstractos. Esto permite el razonamiento arquitectural después del cual es posible refinar las especificaciones de UCM en modelos de escenarios más detallados.

UCM cubre el hueco de modelado entre requisitos y el diseño, y provee un marco de trabajo para especificar el comportamiento. Permite dejar la definición concreta de los componentes de un escenario para fases más apropiadas del proceso de desarrollo.

Una especificación en UCM (de la cual se muestra un ejemplo de su uso en un diagrama en la Figura 3) describe los puntos de entrada y de salida de un escenario en un formato gráfico que trata de ser apropiado para modeladores y personal no especialista en el área debido a su aspecto visual simple y su naturaleza intuitiva a la vez que trata de ser lo suficientemente robusto para soportar el análisis de las características de los requisitos de calidad (verificable, completo, consistente, sin ambigüedades, entendible, trazable y modificable).

UCM permite la transición entre lo informal y lo formal conectando los modelos de objetivos y los requisitos en lenguaje natural y diseño de una manera explícita y visual.

Su principal objetivo es proveer el nivel preciso de formalidad en el momento adecuado del proceso de desarrollo.



**Figura 3** Ejemplo de diagrama UCM [51].

### 2.3 El concepto de valor

El término “valor” es referenciado de forma vaga en multitud de literatura. Mucha de ella deja por sobreentendido un término que es demasiado vago en su definición y en muchas ocasiones depende de interpretaciones subjetivas. En la literatura existen abundantes referencias al “valor” de una organización o de un cliente y cómo conseguir realizar soluciones que contribuyan positivamente a ese valor. En esta sección se repasará la concepción en la literatura de este término, se explicará cómo se utilizará este término en el contexto de este trabajo y su representación en un modelo de valor.

En primer lugar, el término “valor” se encuentra ampliamente utilizado en multitud de publicaciones en la literatura, tanto del ámbito de negocio, como en el de la ingeniería del software y la gestión de proyectos; y como se ha mencionado, es un término que no cuenta con una definición común y que muchas veces se da como un concepto sobreentendido [29]. En [29] se recopilan algunas de las definiciones aparecidas en la literatura:

- Barnett [30]: “... el valor de negocio, como medida de los ingresos del negocio, el precio de las acciones, cuota de mercado u otras métricas de negocio. El Valor está en los ojos del cliente...”
- Patton [31]: “Valor de negocio es algo que entrega beneficios a la organización que paga por el software en la forma de un aumento de los ingresos, evitación de costes o una mejora en el servicio”.
- Pettit [32]: “Valor de negocio es un vehículo de comunicación: usamos el valor de negocio para comunicar valor, prioridades, motivación...”.
- Rawsthorne [33]: “Valor de negocio es por lo que la dirección está dispuesta a pagar; el valor solo puede ser definido por el cliente final. Y solo es significativo cuando se expresa en términos de un producto específico (un bien o un servicio, y a menudo ambos a la vez), que cumple las necesidades de un cliente a un precio concreto en un momento concreto”.
- Poole [34]: “Podría no ser posible definir el valor de negocio de TI independientemente de otras actividades. Qué es valor de negocio:



$$\text{Valor de negocio} = F(x) + F(y) + F(z) + \dots$$

Esto es, una función compleja donde debemos equilibrar múltiples cosas... ¡mientras cambian!

- Valor es un factor de beneficio medible entregado, asociado a un entregable, a un receptor [54].

A pesar de que en [29] se investiga desde la perspectiva ágil, las definiciones son lo suficientemente generales para ser tenidas en cuenta en cualquier ámbito. En el mismo trabajo, se resalta que estas definiciones son de artículos de profesionales, de hecho no se pudo encontrar un artículo científico en el que explicara el valor en un contexto ágil ya que los autores piensan que valor es un concepto evidente por sí mismo ya que ha sido intensamente estudiado por las ciencias económicas.

De estas definiciones se puede concluir que el valor de negocio en la práctica tiende a ser cualitativo, para ver como una práctica crea o no valor hace falta asociarla a una ganancia tangible del negocio, lo cual es problemático. Aunque sí se han encontrado casos en los que se trata el valor de manera cuantitativa, en general se trata de un valor cualitativo.

También, el valor de negocio tiende a ser subjetivo, tanto porque depende del punto de vista del *stakeholder* como porque se expresa de manera informal como “Valor es algo que me hace sentir bien”. Esto es problemático cuando la fuente del valor del cliente, su representante ante el equipo de desarrollo, no representa la totalidad de los stakeholders y guía la priorización de requisitos: los requisitos son priorizados en base a ese valor y según la importancia que estos tengan para el proyecto, pudiendo llegar a situaciones en las que una vista parcial del sistema se impone al resto. Este es un aspecto especialmente relevante en el presente trabajo y que se abordará en los siguientes capítulos.

El valor que entrega la solución TI depende de otros procesos fuera de la propia solución (procesos de negocio internos, de otros stakeholders, otras organizaciones...) por lo que es difícil de definir el valor de negocio independientemente del resto de actividades de negocio no informatizadas, otros sistemas existentes o los stakeholders implicados.

Otro aspecto inferido es que el valor de negocio de la solución de TI requiere un grado de confianza en los indicadores de valor que se presentan ya que pueden no ser precisos.

El valor del cliente puede ser visto desde diversas perspectivas, todas ellas complementarias. El valor del cliente, es decir aquello que el cliente valora y que ayuda a cumplir con sus objetivos y que se impone como una restricción o condición a un proyecto o el valor utilizado en la organización para priorizar entre distintos proyectos puede observarse en términos del negocio o puede estudiarse de manera que dirija los desarrollos de software (a fin de entregar productos con el máximo valor al cliente). A continuación se profundizará en el concepto desde el punto de vista del negocio, de la ingeniería de software y especialmente del desarrollo de software ágil.

### **2.3.1 Perspectiva de negocio**

Jim Highsmith en [35] identificaba el concepto como un término informal que incluye todas las formas de valor que determinan la salud y bienestar de una organización a largo plazo. El valor de negocio expande el concepto de valor de una empresa más allá de lo

económico para incluir otras formas de valor como el valor del empleado, el valor del cliente, el valor del proveedor, el del socio, del aliado, el administrativo y el social; muchos de ellos no medidos directamente en términos económicos.

El valor es importante para centrarse en qué es prioritario para la organización y realizar decisiones de priorización entre las oportunidades de negocio, proyectos y productos posibles. El valor tiene componentes tangibles (financieros) e intangibles ambos críticos para el éxito a largo término.

Una variación en el valor del negocio puede ser originada por una mejora de los procesos empresariales y podría ser medida por factores llamados *Value Drivers* como serían: el aumento de beneficios empresariales, mejora de la rentabilidad empresarial, cuota de mercado, aumento de la liquidez financiera, repuesta de las campañas de marketing, mejora de la fidelidad de los clientes, posición de liderazgo, portafolio de productos, cartera de clientes, mejora de la satisfacción de los empleados, aumento de la capacidad de innovación empresarial, etc.

El valor de negocio es utilizado clásicamente por las empresas para tomar decisiones estratégicas usando por ejemplo técnicas como el *Return Of Investment* (ROI) para determinar qué proyectos son más provechosos para una organización.

Una de las metodologías más utilizadas para medir y gestionar el valor (*business value*) es el *Balanced Scorecard Methodology* (BSC (*Balanced ScoreCard*) / CMI (Cuadro de Mando Integral)) [36]. Aunque el concepto también aparece en *Value Chain* (Cadena de valor) definido por Michael Porter [37] y *Management by Objectives* definido por Peter F. Drucker [38].

La *Balanced Scorecard Methodology* es un modelo de planificación estratégica que basa el desarrollo y la medida de la gestión empresarial en indicadores financieros. Esta metodología trata de alinear la estrategia del cliente con las iniciativas, operaciones (procesos) y presupuestos que implementan esta estrategia, proporcionando medidas estratégicas que permitan evaluar el éxito de la estrategia.

En CMI/BSC se descompone el valor del negocio en cuatro componentes: las proposiciones de valor del cliente, los ingresos, recursos y competencias y procesos.

Una proposición de valor del cliente (*Customer Value Proposition, CVP*) es una oferta que ayuda a los clientes a hacer un trabajo importante de forma más efectiva, conveniente o asequible que las alternativas. La estrategia (de negocio) está basada en una propuesta diferenciada de valor del cliente, la satisfacción del cliente es la fuente de una creación de valor sostenible.

La cadena de valor es una técnica de análisis de negocio [36] basada en la vista de la organización como un sistema, hecho de subsistemas en el que cada uno presenta unos puntos de entrada, realiza unos procesos de transformación y sus resultados aparecen como salidas. Es decir la cadena de valor representa el flujo de valor dentro de los componentes organizativos de la empresa.

Esta técnica es utilizada para la planificación estratégica, dado que la captura de los elementos que intervienen en la cadena permiten razonar sobre las relaciones entre estos elementos y analizarlas en busca de mejoras; por ejemplo localizando dos elementos

adyacentes de la cadena en lugares geográficamente cercanos o ideando nuevos modelos de negocio para eliminar intermediarios.

A pesar de la extensión del estudio del valor en el apartado de negocio, en muchas ocasiones el valor se confunde con las herramientas, algunas de las más populares han sido presentadas en esta sección, y los indicadores que lo representan y es complicado llegar a la fuente que hace que los indicadores (y por tanto el valor) fluctúe. Esto puede ser un detalle menor cuando se toman decisiones estratégicas, dado que a fin de cuentas una acción que mejore los indicadores de valor y por tanto reporte más beneficios a la organización será positiva independientemente de que se conozca la fuente de valor. Sin embargo, si el objetivo es construir software que sea la fuente de ese valor o un canalizador del mismo, el enfoque de las herramientas para localizarlo y expresarlo será distinto.

### **2.3.2 Perspectiva de la ingeniería del software**

Desde la crisis del software de los años 80, la ingeniería del software se ha preocupado por la investigación y uso de aquellas técnicas y métodos que permitan a los desarrolladores crear software útil y correcto para el cliente, dicho de otra manera, entregar software que satisfaga los requisitos de funcionalidad esperados por el cliente y aquellos requisitos no funcionales que permitan el uso satisfactorio del software. Debido a ello, se encuentran menciones al valor (del cliente) en el campo de la ingeniería de software en multitud de ocasiones, aunque muchas veces, como es habitual con el concepto en los otros campos, sin recurrir a ninguna definición estructurada y tratando el término como sobreentendido. Esto no es así, como por ejemplo demuestran en [41] siendo un tema de debate en la comunidad.

Sin embargo, dentro de la ingeniería del software una línea de pensamiento se ha centrado exclusivamente en la postura de que el software debe aportar valor al cliente y este debería ser el objetivo del proceso ingenieril. Este es el objetivo de la propuesta de Boehm (2006) con la creación de la *Value Based Software Engineering* [41] que “lleva las consideraciones de valor al frente para soportar las decisiones de la Ingeniería del Software a todos los niveles para conseguir o reconciliar los objetivos explícitos de los stakeholders involucrados, desde el personal de mercadotecnia y los analistas de negocio a los desarrolladores, arquitectos y expertos de calidad y desde los expertos de procesos y mediciones hasta a los gestores de proyecto y los ejecutivos”.

En ese mismo texto Boehm aclara qué es lo que entiende por valor: valor relativo, utilidad o importancia y se usa como sinónimo de proposición de valor, función de utilidad y condición ganadora (“*value proposition*,” “*utility function*,” and “*win condition*”). Esta perspectiva se pone en contraposición a la definición del diccionario (Webster, 2002) de “valor” como: “el valor monetario de algo: su precio de mercado”, argumentando que esta definición dejaba fuera del valor, elementos de negocio que sin un valor financiero, los expertos señalan que sí contribuye al valor de una organización. El autor argumenta que una definición menos estricta permite la inclusión de situaciones menos rigurosamente analizables pero más amplias como las consideraciones personales, interpersonales o éticas. Jim Highsmith, desde una perspectiva de negocio, coincide con esta postura, como se ha comentado en la sección anterior.

Esta corriente de pensamiento coloca en el foco el valor para el cliente y lo utiliza para guiar la totalidad del proceso de desarrollo desde la captura de requisitos a las fases de pruebas.

En la literatura, el punto del ciclo de vida del software en el cual es más relevante el uso del valor o su discusión es en la fase temprana de la captura de requisitos y también su uso a la hora de priorizar los requisitos. En esta línea se encuentran ejemplos como [42][47] en los cuáles no se hace hincapié en una definición formal pero sí se presentan técnicas y herramientas, formales o informales para identificar el valor e introducirlo en la captura de requisitos y en la priorización de los mismos. De forma general, la forma predilecta de capturar el valor es utilizar métodos cuantitativos o cualitativos relativos que permiten comparar los elementos de valor entre sí y discernir los más valiosos que el resto y realizar una clasificación. Este es el caso del uso de *Analytic Hierarchy Process* [42] en la priorización de requisitos por valor [43], que compara entre pares de elementos para otorgarles un valor fijo.

En el aspecto más estructurado, y quizás más relacionado con la definición exhaustiva del contexto del sistema y la intencionalidad de los involucrados que lo forman, se encuentran los lenguajes de descripción de la intencionalidad o de objetivos como  $i^*$  y GRL (extensamente explicados en la sección 2.5), donde los elementos de valor son capturados como objetivos de los distintos stakeholders, interconectados entre sí según el efecto que tengan en otros, las necesidades de los stakeholders de esos elementos o su descomposición jerárquica.

En una dirección ligeramente similar a los movimientos de negocio e<sup>3</sup>-value y *Value Delivery Modeling Language* (VDML) [54] (también detallados en la sección 2.6 en sus respectivos apartados) tratan de capturar los flujos de valor entre las unidades operacionales, bebiendo directamente de técnicas de negocio como la cadena de valor.

Lo que es evidente es la existencia de la preocupación de capturar los elementos de valor para el cliente y que producirá el sistema de información objetivo de manera que este responda ante estas expectativas y solo así de verdad se estará construyendo software de utilidad para el cliente, independientemente del tipo de representación o técnica utilizada para lograrlo.

#### **2.4.2.1 Perspectiva del desarrollo de software ágil**

Dentro de la literatura de la ingeniería del software, una subsección debe dedicarse al campo del desarrollo de software ágil. En dicho campo, el valor ha estado presente desde la misma concepción. En los principios del manifiesto ágil [4], considerado el punto de partida oficial de la corriente ágil tal y como se conoce en la actualidad, ya aparece una mención al valor en sus 12 Principios Ágiles: “Nuestra máxima prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor”. Esta mención, en un texto que se considera la piedra angular de la corriente de pensamiento ágil, no es tangencial sino que lo considera “máxima prioridad” del desarrollo de software.

En los principios del desarrollo de software ágil se incide en la inclusión de la perspectiva de negocio en el proceso de desarrollo, haciendo que el personal encargado de esa perspectiva se involucre en el proceso de desarrollo, validando diseños y decisiones y aportando conocimiento del dominio y del negocio con el fin último de crear software valioso.

A pesar de que “software valioso” no es un término que se hay concretizado en la literatura se ha tratado de afinar este concepto y trabajar en métodos que permitan conocer cuando el software es valioso y en qué medida se está entregando valor al cliente.

Jim Highsmith propone su “triángulo ágil” [44] en contraposición al triángulo tradicional de acero de los proyectos, cuyas restricciones se basan en tiempo, coste y alcance; por una perspectiva de valor, calidad y restricciones (punto que engloba las preocupaciones de la visión anterior). Este mismo autor propone la valoración del valor entregado mediante la técnica del *Value Points Analysis* (VPA, Análisis por puntos de Valor).

En el trabajo de *Estimating, Planing and managing Agile Web development projects under a value-based perspective* [45], los autores proponen el uso de esta técnica y de Earned Value Management (EVM) para la gestión de un proyecto ágil, haciendo principal incidencia en el uso del valor para la planificación de las iteraciones del proceso de desarrollo.

Se propone AgileEVM [37] como implantación del método clásico *Earned Value Management* en proyectos ágiles, concretamente aquellos que implementen Scrum. Este método trata de integrar el alcance, la planificación y los recursos y medir el desempeño del proyecto. Esta técnica se utiliza a nivel de planificación de proyecto pero en el trabajo mencionado muestran cómo integrarla con Scrum y proponen métricas y técnicas para estimar el valor de las historias de usuario y la contribución al valor total que se está realizando durante el proyecto.

Las técnicas revisadas tienen en común la intención de su uso: conseguir establecer una valoración del valor de cada requisito de manera que se pueda establecer un orden de prioridad basado en este criterio. Los métodos para capturarlo pueden diferir pero la mentalidad ágil hace que todas las propuestas vayan en esta dirección por lo que tienden a incorporarse como prácticas en un desarrollo iterativo, ágil y orientado a satisfacer el valor del cliente en cada iteración. En definitiva, a incorporar las preocupaciones del cliente a un producto software de forma ágil.

## **2.4 Alineamiento entre IT y negocio: El valor como elemento cohesionador**

El objetivo final de una solución TIC debe ser la satisfacción de los objetivos de la organización. Esto solamente se alcanza cuando se consigue el alineamiento estratégico entre TIC y los objetivos, actividades y procesos del negocio de manera que se encuentren en armonía [47].

Se ha demostrado que un alineamiento estratégico influencia de manera positiva la efectividad de las soluciones TIC por lo que la alineación de TIC y estrategia de negocio ocupa los puestos de alta prioridad para los directivos y ejecutivos TIC [47]. A pesar de ello y como se ha comentado en este capítulo, este hecho es ignorado en numerosas ocasiones por las técnicas de la ingeniería de requisitos, más centradas en la descripción de la operatividad y que confían en la experiencia de los desarrolladores a la hora de identificar requisitos y diseñar productos adecuados a las necesidades del cliente.

Varios aspectos del análisis de negocio se han incluido como se indica en [47] en la investigación de la ingeniería de requisitos. Algunos aspectos como la estructura

organizacional y las relaciones de dependencia entre actores en un sistema, el análisis de valor económico y de negocio, el modelado de proceso de negocios dirigido por los objetivos de la organización y la elicitación de objetivos organizacionales de los cuales derivar requisitos tratan de establecer el puente la perspectiva de negocio y la de IT de manera que sus resultados alineen los objetivos de negocio y el sistema de información objetivo. Un ejemplo de estas técnicas es [47] donde se propone un acercamiento analítico para la verificación y validación de los requisitos en relación con su alineación con la estrategia de negocio utilizando el modelado de objetivos para representar la estrategia de negocio como requisitos. Queda patente que para construir sistemas TIC de valor, los técnicos primero deben situarlo en el contexto de negocio apropiado, no solamente comprender el dominio y los requisitos que el software debe poseer, si no que el conocimiento ha de ser más profundo. Esto significa que se deben conocer y comprender el modelo de negocio que debe soportar el sistema, las intenciones y metas de la organización que motivan la creación de valor y de un sistema de información que lo soporte. La historia reciente de los negocios electrónicos demuestra que fallar a la hora de entender el modelo de negocio a menudo resulta en negocios de poca duración o incluso en la quiebra [48].

Es necesario, por tanto, contemplar durante todo el proceso de desarrollo de software el objetivo general del desarrollo: que este sirva para que el negocio cumpla con sus objetivos de valor.

## **2.5 El Modelado de valor**

En el campo del desarrollo dirigido por modelos se ha abordado el modelado del valor desde diversas perspectivas. En el campo de la ingeniería de modelos, se incluyen propuestas para el modelado explícito del valor (e<sup>3</sup>-value, Value Delivery Metamodel) en los que se hace referencia a una acepción concreta de valor y se trata de plasmar sobre los modelos; y otras del ámbito de la descripción de la intencionalidad y el modelado de objetivos (i\*, *Goal Requirements Language*) cuya semántica incluye a las acepciones de valor.

En esta sección se presentarán y analizarán las distintas propuestas halladas en la literatura y se seleccionará la más indicada para representar el valor en la metodología propuesta (capítulo 3).

### **2.5.1 i\***

i\* es una metodología de desarrollo de software soportada por modelos orientadas a objetivos. Los modelos i\* representan los objetivos de una organización y sus interacciones con otras organizaciones o stakeholders. El espacio de trabajo de i\* propone un acercamiento orientado a agentes a la ingeniería de requisitos centrándose en las características intencionales del agente.

i\* fue concebido por Eric S. Yu en su tesis doctoral en 1995 y publicado en [8] por primera vez.

El aproximamiento al modelado de i\* es un intento de traer el entendimiento del contexto social a los procesos de la ingeniería de sistemas introduciendo conceptos sociales seleccionados en el núcleo de la actividad diaria de los analistas de sistemas y diseñadores, proponiendo una ontología social para los constructos de modelado principales.

El lenguaje de modelado de i\* se centra en las propiedades de intención en vez del comportamiento central del sistema. Los lenguajes de modelado tienen tendencia a centrarse en el *¿cómo?* mientras que i\* trata de definir el *porqué*, explicar y caracterizar el comportamiento de los diferentes involucrados con el sistema. El término “i\*” significa intencionalidad distribuida.

El modelo de i\* trata de representar que no existe una sola visión del problema sino que trata de modelar cómo los distintos involucrados tratan de buscar el avance de sus intereses considerando como influyen en ellos diferentes configuraciones de las relaciones de dependencia.

En el *framework* de i\* existen dos tipos de modelos:

- **Strategic Dependency (SD) model (Modelo de dependencia estratégica):** Expresa una red de relaciones de dependencias directas entre actores. Un enlace entre actores indica que un actor (*dependor*) depende de otro (*dependee*) por un elemento (*dependum*). El *dependor* quiere que el *dependee* haga la aserción correcta, el *dependum*, sin especificar cómo se conseguirá. Esto da libertad al *dependee* para conseguir sus objetivos sin tener que modelar con detalle cómo será conseguidos cuando esto no depende de este actor.

Este modelo se puede contrastar con los modelos de proceso empleando ontologías dinámicas. Los modelos de proceso se centran en el flujo de control o de información; el modelo SD tiene un nivel de abstracción más alto y se centra en describir qué es lo que los actores quieren de cada uno.

Este modelo reconoce la autonomía de los actores en el mundo social, de manera que estos solo interactúan por mediación de las dependencias. Estas relaciones externas entre actores son representadas, pero no descritas en detalle operacional.

- **Strategic Rationale (SR) model (Modelo de racionalidad estratégica):** En este modelo se atribuyen objetivos, tareas, recursos y objetivos soft a cada actor, esta vez como elementos intencionales internos que el actor quiere conseguir, dentro de sus fronteras. Es decir, se representan los elementos internos relevantes para el actor, los objetivos que quiere lograr y las tareas relevantes junto con los recursos que consumen o generan dichas tareas y sus relaciones con los elementos fuera de las fronteras del actor.

En ambos modelos unos actores (stakeholders del contexto del sistema) establecen enlaces de dependencia sobre elementos intencionales (tareas, objetivos, objetivos soft) que son relevantes para el actor. Estas dependencias pueden variar en grado de fuerza: *abierta*, si la dependencia falla el *dependor* no será significativamente afectado; *involucrado*, si la dependencia falla el *dependor* será significativamente afectado; o *crítica* si la dependencia falla el *dependor* no será capaz de satisfacer el objetivo asociado.

Así mismo, la definición de los elementos y las tareas pueden jerarquizarse por medio de enlaces de manera que se creen sub-tareas o sub-objetivos o se representen alternativas para la satisfacción de determinadas dependencias. A su vez, se pueden representar enlaces de contribución entre los diferentes elementos para expresar en qué grado (positiva o negativamente) afectan unos elementos intencionales a otros.

i\* hace especial hincapié en la representación de alternativas de manera que el análisis del modelo permita identificar oportunidades o debilidades y seleccionar aquellas alternativas más válidas.

A pesar del tiempo que i\* lleva incorporado a la literatura, no cuenta con una especificación o un estándar formal sobre el lenguaje y su uso, aunque sí que existe un fuerte esfuerzo investigador se ha observado que es habitual la extensión o limitación de las distintas propiedades del lenguaje de forma ad-hoc para las necesidades concretas de un proyecto. La semi-formalidad del lenguaje flexibiliza esta posibilidad pero por contrapartida dificulta la interoperabilidad entre distintos modelos o herramientas para este lenguaje de modelado así como dificulta su aprendizaje. Aunque el lenguaje sí que i\* cuenta con herramientas que soportan la creación de modelos y su mantenimiento como OpenOME [50].

En la literatura se han encontrado algunas limitaciones a este lenguaje. En primer lugar, para soportar las decisiones ante alternativas representadas en el modelo se necesitaría de elementos capaces de representar las creencias, asunciones y justificaciones (*Beliefs, assumptions, justifications*) a pesar de que este papel puede ser representado por los objetivos soft, todavía estos conceptos no han sido integrados por completo. En segundo lugar, la representación del cambio en i\* es limitado. Las configuraciones de alternativas o modelos históricos (*as-is, to-be, could-be*) se representan en modelos SD separados. Mientras que un modelo SR puede mostrar múltiples alternativas y el detalle de cómo estas contribuyen a distintos stakeholders, este nivel de detalle también limita la presentación ya que la complejidad visual del modelo crece. Por ello, los cambios en el entorno, sobre todo si son continuos y graduales son difíciles de expresar. Por último, los modelos de i\*, en especial el “modelo SR” detallado pueden ser demasiado pesados, complejos visualmente y con demasiados conceptos y sutilezas que, mejorando su expresividad, pueden resultar complejos para el personal no técnico o técnicos novatos en el uso del lenguaje.

Sin embargo, la propuesta intencional de i\* permite la expresión de cualquier tipo de concepto de valor y permite describir de forma sencilla cualquier tipo de elemento de valor y sus relaciones entre sí y con los actores. A diferencia de otras propuestas centradas en el modelado de requisitos centradas en la funcionalidad, este lenguaje es capaz de modelar y analizar el interés de los stakeholders.

De la misma manera, i\* es un lenguaje de modelado pensado para la fase temprana de la ingeniería de requisitos mientras que otros frameworks que proponen la representación del conocimiento como soporte a la ingeniería de requisitos no distinguen las fases temprana y madura.

### **2.5.2 Goal-oriented Requirements Language**

*Goal-oriented Requirements Language* (GRL, Lenguaje de requisitos orientado a objetivos) forma parte del estándar de la International Telecommunications Union (ITU) [52] *User Requirements Notation* (URN, Notación de Requisitos de usuario) [23] desde el año 2009. GRL es un lenguaje para el modelado y el análisis de requisitos, basado en objetivos, escenarios y los enlaces entre ellos. Su estructura es semi-formal y ligera. El estándar URN está formado por GRL así como el *framework* de *Non-Functional Requirements* (Requisitos no funcionales, NFR) y *Use Case Maps* (Mapas de casos de uso, UCM).



GRL surge a partir del lenguaje de modelado de la intencionalidad  $i^*$  del cual hereda parte de su notación, y semántica pero no de manera total y optando en muchas ocasiones por otras aproximaciones. Notable es la ausencia de diagramas tipo como el SR y SD de  $i^*$ . El estándar de GRL se limita a la creación y descripción de un metamodelo a la vez de dotar a los elementos de la semántica apropiada para la descripción de la intencionalidad, pero no da recomendaciones acerca de los procesos para construir sus diagramas.

Esta notación soporta la elicitación, análisis, especificación y validación de requerimientos desde la perspectiva de la intencionalidad. Permite a los ingenieros de requisitos descubrir y especificar requerimientos para un sistema propuesto o evolucionado y analiza estos requisitos para la corrección y la completitud. Combina la orientación a los objetivos y otros conceptos intencionales (requisitos no funcionales, atributos de calidad y razonamientos sobre alternativas y *tradeoffs*) y los aspectos de caso de uso, requisitos operacionales, requisitos funcionales y razonamiento sobre el rendimiento y la arquitectura.

GRL permite tratar con la elicitación temprana de requisitos, la fase en la cual deben identificarse los elementos de valor. Así mismo posee una comunidad investigadora activa, con objetivos marcados para los próximos años [53] y posee una herramienta bastante madura basada en Eclipse (jUCMNav).

El lenguaje además de poseer los elementos descriptivos posee herramientas de análisis propias para evaluar los modelos de GRL de manera que sea posible detectar alternativas y/o errores de diseño.

Un diagrama de GRL muestra a un conjunto de actores, los stakeholders del sistema, relacionados con elementos intencionales (tareas, objetivos, objetivos soft, recursos... sobre los que los actores poseen algún interés intencional) y las relaciones entre ellos de manera que representen el contexto intencional de un sistema de información, incluyendo información de cualquier tipo que sea de valor para los stakeholders.

### **2.5.3 E<sup>3</sup>-value**

Presentado en [9] e<sup>3</sup>-value es un lenguaje informal para la representación del valor económico. Un diagrama e<sup>3</sup>-value describe los flujos de valor, en este caso limitado a los intercambios económicos.

E<sup>3</sup>-value se propone como un método de exploración para la primera fase de la ingeniería de requisitos basada en el valor como una forma de entender y articular una idea de producto. Es una aproximación ligera, debido a los cortos tiempos de desarrollo y a los pocos recursos humanos empleados en la fase de exploración.

E<sup>3</sup>-value tiene en cuenta tres puntos de vista en el desarrollo de software: el *punto de vista del valor*, centrado en la forma de creación de valor económico, distribución y consumo del sistema; el *punto de vista del proceso de negocio*, centrado en los procesos necesarios para poner en práctica una nueva proposición de valor y se centra en quién debe llevar a cabo esos procesos; y por último el *punto de vista del sistema de información* el cual se centra en constituir, en granularidad gruesa, componentes de un sistema de información para ser desarrollado.

El lenguaje de e<sup>3</sup>-value posee una sintaxis gráfica y un acercamiento basado en escenarios, escenarios operativos que describen el comportamiento del sistema, y escenarios evolutivos que representan posibles eventos en la vida del sistema que podrían provocar cambios.

En el modelo de valor de e<sup>3</sup>-value se representan los distintos actores del escenario que están intercambiando valor económico entre sí y cómo lo están haciendo.

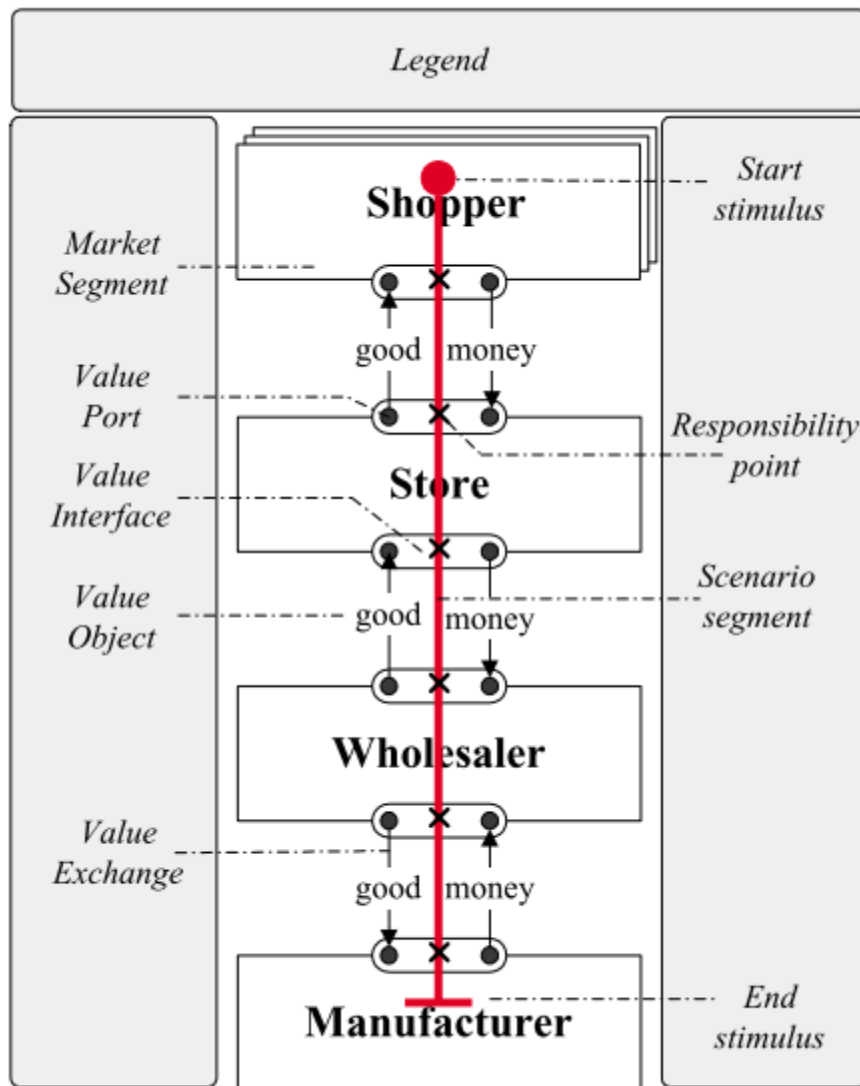


Figura 4 Ejemplo anotado de diagrama e<sup>3</sup>-value [9]

Como se muestra en la Figura 4, un actor (*shopper, store, wholesaler...*) es una entidad económicamente independiente (con respecto a la creación o consumo de valor) que intercambian objetos de valor (marcados como *value object*), los cuáles son bienes, servicios, dinero o experiencias del consumidor que son de valor para alguno de los actores.

Los actores ofrecen sus ofertas de valor (*Value Offering*) a través de sus interfaces de valor (*Value Interface*) representando cada una de ellas en un puerto de valor (*Value*

*Port*). Las actividades de valor son representadas en el interior de los actores y en ellas se describe las operaciones que realiza un actor para conseguir o consumir valor.

Para la descripción de la operacionalidad de los intercambios de valor se utiliza la técnica *Use Case Maps* (incluida en el estándar URN al cual también pertenece GRL).

#### **2.5.4 Value Delivery Modelling Language**

*Value Delivery Modelling Language* (VDML) [54] es un estándar del OMG (*Object Management Group*) cuya primera versión apareció en octubre de 2015. VDML es un lenguaje de modelado y de diseño operativo de una empresa centrado en la creación y el intercambio de valor.

En 2009 OMG lanzó la propuesta de desarrollo de un estándar para el modelado del valor, el objetivo de esta era la de integrar los diferentes modelos de valor existentes y dar una visión del conjunto de la lógica de negocio de una organización.

Como estándar de la OMG el metamodelo de VDML cumple con las características del metamodelo de *Meta-Object Facility* (MOF), estándar ampliamente aceptado en la ingeniería dirigida por modelos, lo cual hace que sea fácil de integrar con otras herramientas de la ingeniería de modelos.

VDML integra siete modelos de valor existentes de los ámbitos de negocio y de la ingeniería de software: la cadena de valor de Porter [37], la ontología de modelado de negocio de Osterwalder [55], el análisis de la red de valor (*Value Network Analysis*)[56], el análisis de e<sup>3</sup>-value, el análisis de Recurso-Evento-Agente (*Resource-Event-Agent*, REA)[57], la asociación de la corriente de valor (*Value Stream Mapping*) y el análisis de la arquitectura de negocio orientada a servicios.

VDML está diseñado para soportar la optimización del valor de los stakeholders tanto con respecto a sus actividades de negocio internas y como externas. VDML soporta la medición del valor tanto desde la perspectiva operacional como la de la satisfacción del receptor. Soporta el modelado del valor de negocio intangible y tangible y es compatible con frameworks de gestión de recursos.

También VDML permite modelar las relaciones de negocio colaborativas y las redes de negocios basadas en roles. Al nivel de interacciones entre los socios de negocio, VDML puede representar los intercambios netos de valor entre entidades de negocio o ir en detalle en los intercambios transaccionales para alcanzar acuerdos de intercambio y manejar el intercambio de recursos económicos. Estos modelos soportan el análisis del efecto total de los intercambios entre varias organizaciones, permitiendo a los participantes darse cuenta de las ganancias netas de mantener una relación.

VDML soporta la perspectiva del diseño dirigido por el valor centrándose en las actividades y el flujo de entregables que producen productos o servicios que entregan valor. A su vez puede ser un espacio de trabajo para generar requisitos para el diseño de proceso de negocio pero provee una vista que es diferente de la de las herramientas de modelado de procesos, ya que VDML se centra en el consumo y la producción de entregables y la producción de valor y el rendimiento de las actividades. Todo ello sin entrar en el detalle del control operacional de los aspectos de proceso de negocio; se centra en la entrega de valor y los medios para ese fin.

Según la OMG, VDML es escalable desde actividades operacionales clave a modelos de negocio a nivel industrial y grandes redes de negocios, cubriendo desde la concepción del producto a la comercialización, entrega y soporte al cliente.

Permite a su vez, el análisis estadístico de las diferentes unidades del modelo y diferentes medidas y escenarios. Esta especificación incluida en el estándar de 2015 incluye una notación limitada, normativa y gráfica pero que se espera que evolucione según los usuarios identifiquen nuevas maneras de utilizarla o realicen implementaciones no normativas. A su vez se espera que a las descripciones gráficas sean complementadas con representaciones tabulares o similares.

El uso esperado de esta notación es que sea aplicado para tratar problemas específicos pero que a su vez sea posible su mantenimiento y crecimiento para proveer una abstracción de la operación de una empresa sostenible e integrada.

VDML incluye conceptos de actividades, roles, flujos y participantes (similares a BPMN) pero a un nivel más alto de abstracción de la actividad de negocio para centrarse en las características estadísticas de las actividades, recursos, entregables y contribuciones de valor que proveen una perspectiva empresarial de la operación de negocio.

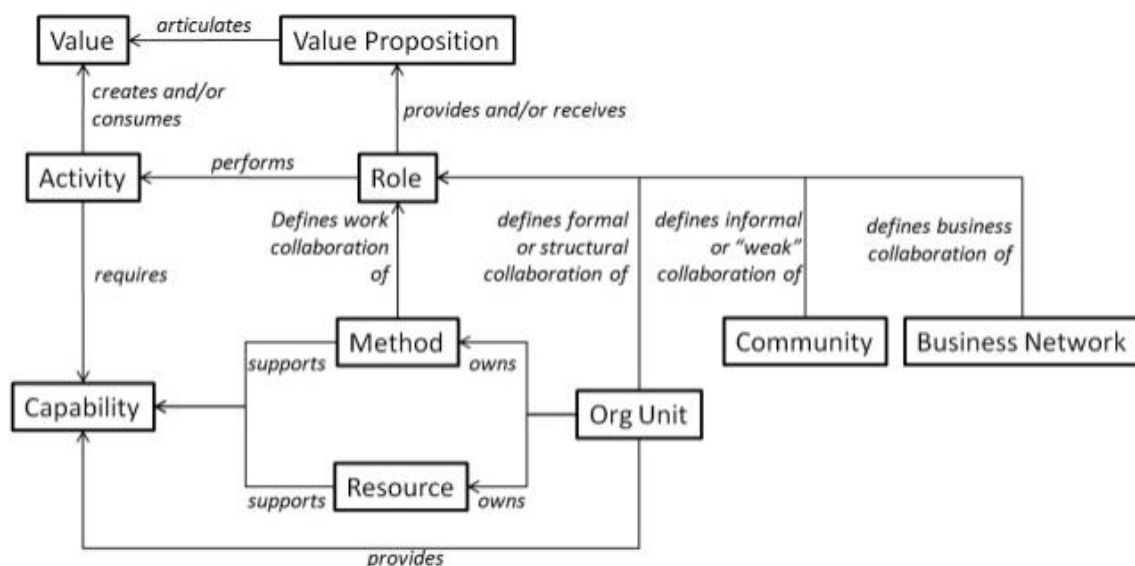
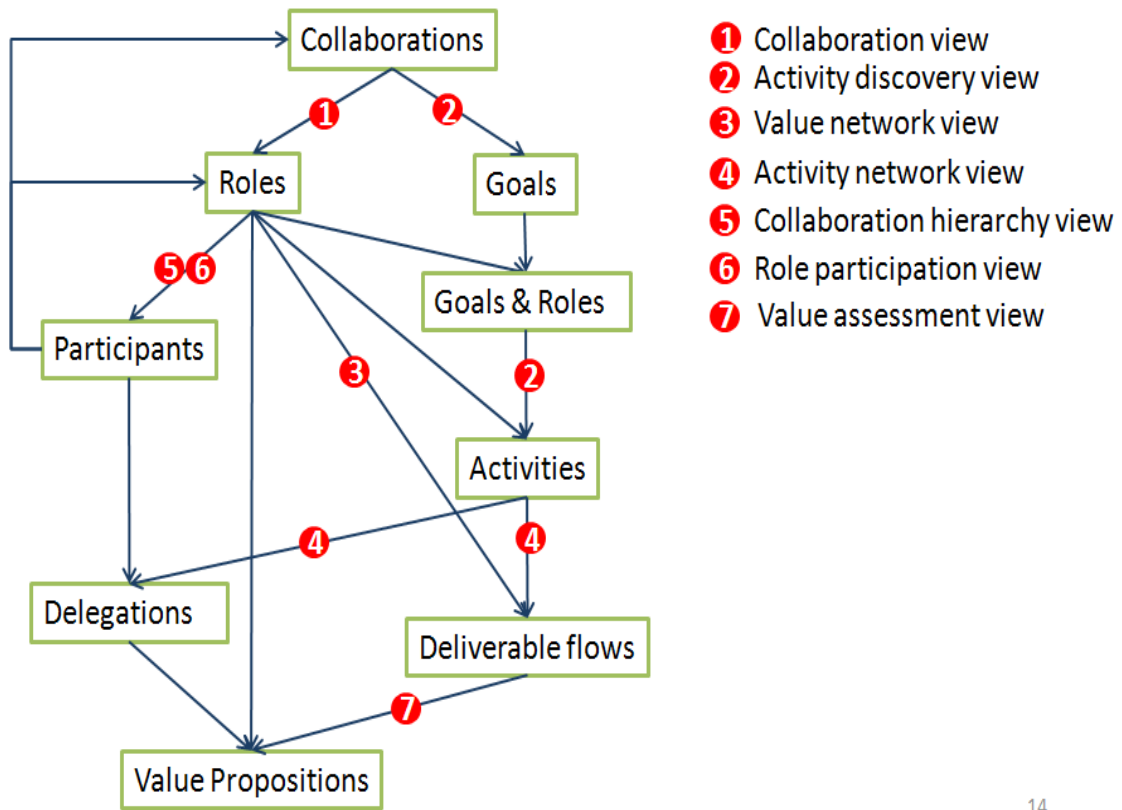


Figura 5 Metamodelo simplificado de VDML [59].

Un modelo VDML puede tener múltiples escenarios, cada uno pudiendo tener diferentes medidas asociadas con los elementos del modelo, representando diferentes circunstancias de operación y configuraciones. Estas medidas pueden ser introducidas por el usuario o calculadas de forma automática.

La Figura 5 es una simplificación del metamodelo de VDML en la cual se explican las diferentes relaciones entre los elementos de VDML. Se puede ver como las redes de negocio, las comunidades y las unidades organizacionales definen colaboraciones de roles. Estos roles proveen o reciben las proposiciones de valor que articulan el valor. A su vez un rol puede realizar una actividad que crea o consume valor. Las unidades organizacionales poseen recursos y métodos que soportan las capacidades necesarias para desempeñar una actividad.



14

Figura 6 Vistas y elementos relacionados en VDML [60].

Como se puede observar en la Figura 6, existen diversas vistas de un diagrama VDML. A continuación se recogen algunos ejemplos de estos tipos de diagramas tomados de [60]

La vista de colaboración (1) es similar a las conversaciones de BPMN 2.0. Los octógonos representan colaboraciones, los círculos representan actividades de colaboración con roles. Esta vista es la adecuada para repasar el ecosistema más amplio de múltiples colaboraciones relacionadas.

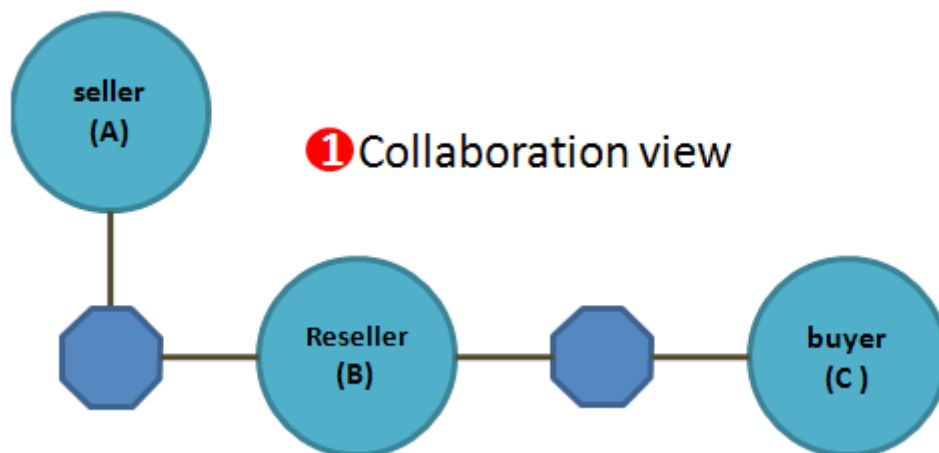


Figura 7 Vista de colaboración.

Cuando los octógonos de colaboración se amplían, el resultado es una vista de red de valor (3), encargada de mostrar los flujos de entregables entre los roles en las

colaboraciones. Como se puede ver en la figura, los roles se resaltan y las actividades están ocultas.

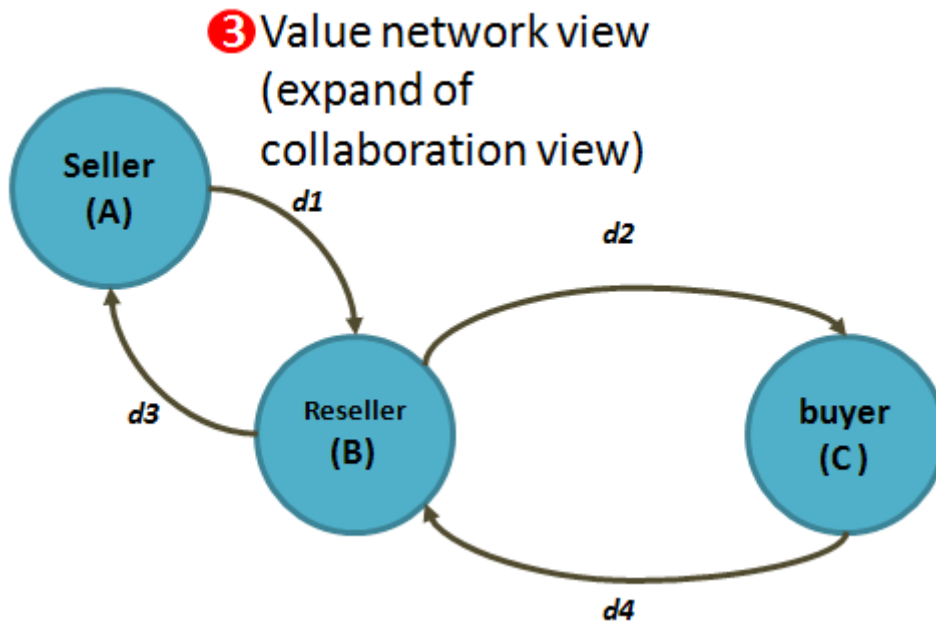


Figura 8 Vista de red de valor.

Para mostrar estas actividades se utiliza la vista de red de actividades (4). Los roles pueden representarse como calles de una piscina.

#### 4 Activity network view

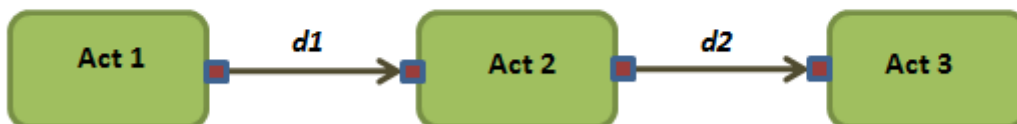


Figura 9 Vista de red de actividades

También puede utilizarse la vista de la jerarquía de colaboración para representar las distintas partes de la organización (5).

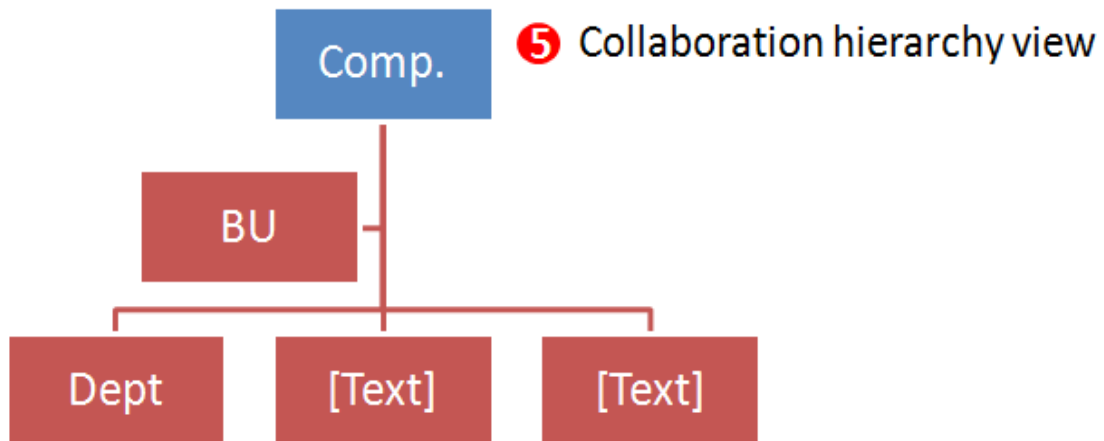


Figura 10 Vista de la jerarquía de colaboración.

Por último, la siguiente vista tabular de la participación de los roles permite observar que participantes realizan qué roles.

Participant	Participates in				
	Engineer in Team X	Member p of Project P			
Employee A	√	√			
Team T			√	√	

**6 Role participation view**

Figura 11 Vista de la participación de roles.

**2.6 La priorización de requisitos dirigida por el valor**

La priorización de requisitos es un problema recurrente en la ingeniería del software en particular y la gestión de proyectos en general. La priorización es el proceso por el cual se ordena un conjunto de elementos según un criterio establecido, de manera que los elementos en los primeros puestos sean los más relevantes. En el presente caso, este criterio es el valor. De manera que los elementos más valiosos serán lo que tengan más prioridad para su inclusión en el proyecto debido a que estos aportan más valor en relación al resto.

En la literatura se encuentran multitud de técnicas de priorización que difieren únicamente en la aplicación de los criterios seleccionados para valorar la lista. Estos criterios pueden tener en cuenta diferentes características de los elementos siendo las más destacadas el coste, el valor o importancia y el riesgo. En esta sección se repasarán

alguna de las técnicas halladas para la estimación de la importancia y el coste y se discutirá la selección de estas técnicas y su implementación en el proceso de desarrollo.

### **2.6.1 Estimación de importancia**

La estimación de la importancia de un requisito es un tipo de estimación difícil de objetivar ya que la valoración de la importancia de un requisito depende de la perspectiva de los stakeholders, la experiencia de los mismos, e incluso puede variar dentro de las diferentes unidades dentro de la organización (pueden existir diferentes grupos dentro de la organización que asignarán un nivel de importancia diferente al mismo elemento debido a que este es más relevante según la posición que ocupan en la organización).

Aunque si es deseable una estimación de la importancia precisa, no necesita ser demasiado exhaustiva, ya que los tiempos del proceso de desarrollo son cortos y aunque de gran importancia para el proyecto, los riesgos por una estimación de la importancia de grano grueso son menores. Por ejemplo, al asignar un elemento en el quinto puesto y otro en el noveno puede no tener ningún efecto en el desarrollo ya que ambas se incluyen en la misma iteración. Por ello es común optar por establecer valores de importancia relativos, de manera que ayuda a determinar qué unidades del sistema tienen más importancia para el cliente, pudiendo priorizar entre ellas, sin establecer un método para especificar la importancia de manera mucho más precisa.

La priorización puede establecerse a nivel de requisitos funcionales pero también a nivel de requisitos de calidad. Siendo el primero más común, en este último sentido en [61] se trata de descubrir cómo funciona la priorización de requisitos de calidad en la práctica en once compañías diferentes. Este estudio resalta que no existe un método en la práctica como los encontrados en la literatura y que las organizaciones tienden a priorizar entre los requisitos de manera *ad-hoc*.

A esta práctica le seguían otras técnicas de priorización, también comunes en la priorización de requisitos, como la asignación numérica, el método más tradicional y común basado en agrupar en diferentes categorías los requisitos, como las comunes “alto, medio y bajo”; las comparaciones entre pares como el *Analytical Hierarchy Process* (AHP), una técnica relativa en la que se comparan todos los elementos entre sí para signar el valor relativo entre cada par comparado; aproximamiento coste-valor, en el cual se compara en una gráfica de dos dimensiones el valor de los requisitos y su coste, de manera que cuentan con más importancia aquellos que tengan menor coste y mayor valor; el voto cumulativo (como la prueba de los 100\$), consiste en la distribución de un valor fijo de puntos entre todos los requisitos de manera que los más valiosos reciban más puntos; hasta la más simple de ordenación en la que los requisitos se ordenan de mayor a menor importancia.

La mayoría de técnicas de priorización encontradas pueden ser clasificadas en uno o más de estos tipos ya que se tratan de variaciones o procesos diferentes sobre este mismo tipo de bases.

En [62] se busca el estado del arte de las técnicas de priorización de requisitos encontrándose 49 técnicas de priorización siendo la más referenciada la ya mencionada AHP. En este trabajo también se establece una taxonomía de las escalas de las metodologías (escala ordinal, nominal, de intervalo, ratio).



Las técnicas que se encuentran clasificadas en estas taxonomías ordenan los requisitos en grupos o sub-grupos y números o inscripciones que son asignadas a través de todos los requisitos para reflejar su importancia relativa por lo que ningún requisito se espera a que pertenezca a más de un grupo o subgrupo al mismo tiempo.

En consecuencia, las frecuencias a través de estos grupos son calculadas y el requisito con la frecuencia más alta es tomado como el principal.

También en este trabajo, se describen los procesos utilizados en ellas que van desde la captura de requisitos, al ordenamiento, la exclusión, la aplicación de criterios de valoración, entrenar a los stakeholders, el agrupamiento, la representación de resultados finales etc.

Así mismo, en [62] también se encuentran limitaciones de los métodos existentes que se deben tener en cuenta para su selección y aplicación en la presente propuesta:

- **Escalabilidad:** Uno de los mayores escollos es la escalabilidad de los métodos, la complejidad computacional (tanto de los métodos automatizados como de los no automatizados) es un escollo para su implementación en proyectos de tamaño mediano o grande, siendo AHP el más lento y *planning game* el más rápido.
- **Gestión de cambios:** Los cambios en la valoración de un elemento o su ordenamiento son difíciles de incluir de forma sencilla y sin recurrir a volver a aplicar el método a la totalidad de los elementos.
- **Expresividad:** la mayoría de las técnicas no son aptas para comunicar los resultados con los stakeholders formados por personal no técnico.
- **Dependencias entre requisitos:** pocas técnicas permiten trazar dependencias entre varios elementos a valorar
- **Poco precisas:** la mayoría de las técnicas tienden a tener errores, debido a que las reglas que gobiernan los procesos de priorización no son lo suficientemente robustas.
- **No implementadas en la práctica:** la mayoría de las técnicas revisadas no han sido implementadas para escenarios reales, se señala que probablemente debido a las complejidades asociadas con las priorizaciones y el tiempo requerido para generar requisitos priorizados.

Por otra parte, la priorización es uno de los componentes críticos en las prácticas ágiles, ya que como se ha visto anteriormente, su foco es entregar valor al cliente desde las primeras iteraciones por lo que seleccionar qué requisitos son de mayor valor para el cliente es un apartado crucial. Sin embargo, en la literatura cuesta encontrar ejemplos de cómo la priorización tiene lugar en la práctica [63]. En el estudio de Racheva *et al.* [63] sobre la priorización de requisitos en la práctica tratan de encontrar qué se refiere como valor y cuál es el proceso que en la industria se emplea para estimar este valor o importancia de cada requisito. En este estudio se refleja que la forma de valorar los requisitos, como en el anterior trabajo, suele ser *ad-hoc*, poco rigurosa y muchas veces limitada por los problemas para llegar a un acuerdo por parte de los diferentes stakeholders sobre la importancia de un ítem.

Un aspecto común entre las organizaciones investigadas es que la técnica de priorización emplee un criterio de coste/beneficio. Se descubrió en estas organizaciones que gran parte del esfuerzo para establecer la prioridad entre tareas lo llevaban a cabo los

desarrolladores, mientras que los clientes se limitaban a validar las priorizaciones llevadas a cabo por estos, muchas veces siendo un solo representante de la organización cliente el que decidía sobre la validez de la estimación y por tanto aportando una visión sesgada de las necesidades reales de la organización.

De igual manera, el cliente suele tener problemas para involucrar a su personal con el proceso de desarrollo por falta de recursos, personal necesario para ayudar a establecer las prioridades entre otras tareas. También se resalta que en muchas ocasiones, a pesar de conseguir los recursos para colaborar en el proyecto, los clientes no eran capaces de expresar sus necesidades, incluso llegando a afirmar que no existe un valor objetivo que sea posible tomar como entrada para el proceso de priorización.

No obstante sí que existen técnicas de priorización, técnicas como el *Return of Investment* (ROI), que mide cuánto dinero se obtendrá de una inversión, se utilizan para estos mismos fines en proyectos tradicionales con éxito a nivel de proyecto. Sin embargo, en el caso de proyectos ágiles donde no es posible el uso de estas técnicas a nivel de proyecto: por sus costes, porque requieren de una gran cantidad de asunciones o porque ciertas actividades de valor para el cliente no tienen coste monetario.

En contraposición a este tipo de técnicas Jim Highsmith [44] propone una técnica llamada Análisis por Puntos de Valor (*Value Points Analysis*, VPA). Esta técnica consiste en valorar los ítems que puedan generar valor al cliente una vez implementadas o incluidas en el sistema de información, en el ejemplo de aplicación original, estos eran las historias de usuarios, con valores de una escala finita como la Secuencia de Fibonacci.

Establecer el valor relativo no solamente ayuda a la priorización de tareas sino que también es útil para estimar el esfuerzo necesario que invertir en un conjunto de ítems de valor para el usuario, si el valor es reducido, el coste de llevarlo a cabo también debería serlo ya que no es relevante o no tan relevante para el cliente como otro ítem de mayor valor en el que se debería invertir más esfuerzo si fuera necesario.

La aplicación de la técnica consiste en escoger de entre un conjunto de valores discretos el valor que representa la importancia o relevancia para el valor del cliente que el elemento intencional valorado se cumpla, combinando las técnicas de numeración y agrupamiento.

## **2.6.2 Estimación de esfuerzo**

La estimación del esfuerzo es una técnica utilizada en el contexto del proceso de desarrollo cuyo objetivo es obtener una evaluación, lo más acertada posible de la cantidad de trabajo necesaria para desarrollar una unidad de trabajo (tarea, requisito, historia de usuario...).

Diferentes técnicas de estimación se han propuesto para esta tarea en la gestión de proyectos, siendo un tema recurrente no solamente en la ingeniería del software. Nótese que existe un solapamiento entre las técnicas para la estimación de la importancia y la del esfuerzo, sobre todo en las aplicadas en los métodos ágiles, esto es debido a que ambas técnicas se basan en los mismos principios para la estimación (por ejemplo, basar la estimación en las experiencias previas o el conocimiento del dominio) para realizar la tarea.

A continuación se citan algunas de las más utilizadas [64]:

**-El Juicio de Expertos:** en esta técnica se realiza la estimación en base a los conocimientos y experiencias previas de personas que han realizado un trabajo igual o semejante al cual se quiere determinar el esfuerzo. Este grupo de expertos determina el coste de cada una de las unidades de trabajo mediante consenso. Muchas veces se ‘abusa’ de esta técnica, debido a la falta de datos cuantitativos de proyectos anteriores,

**-Estimación por analogía (Top-Down):** se cuestiona el coste desde los elementos más generales a los más específicos. Es utilizada cuando se cuenta con experiencia en proyectos anteriores, análogos o similares, que pueden servir de referencia. Es una técnica menos costosa y más rápida pero tiene como desventaja que es menos exacta y que se necesita de experiencia y documentación.

**-Estimación paramétrica:** consiste en una estimación con base a parámetros, aunque también puede utilizar datos de proyectos anteriores y datos de referencia. Su principal característica es que la estimación se realiza con base a la relación entre variables, por ejemplo, coste por cantidades producidas, horas hombre por desarrollo, transporte por recorrido, etc.

**-Estimación ascendente (Bottom-Up):** este método de estimación asciende desde la estimación del detalle de cada elemento hacia el elemento más general del objetivo del proyecto. Es una técnica costosa ya que hay que examinar todos los elementos del proyecto en su último detalle para poder abstraer la estimación de los elementos generales. Trabajar a este nivel de detalle permite ser más preciso en las estimaciones.

**-Estimación con tres valores:** también llamada de tres puntos, derivada de la Técnica de Revisión y Evaluación de Proyecto, conocida como método PERT (*Project Evaluation and Review Techniques*), consiste en identificar tres posibles valores, el optimista, el pesimista y el más probable, para llegar a un único valor aproximado. Esta técnica es utilizada en escenarios inciertos y con diferentes antecedentes o muchas variables que puedan afectar el valor final. Existen 2 fórmulas principales, la primera basada en una distribución triangular que en forma de promedio toma la sumatoria del valor optimista, el pesimista y el más probable y lo divide entre 3. La segunda fórmula está basada en una Distribución Beta y es como se muestra a continuación:

$$\text{Coste Estimado} = (\text{Optimista} + 4 (\text{Más Probable}) + \text{Pesimista}) / 6$$

**- El análisis de la reserva:** es una técnica complementaria a la estimación del coste, permite, basado en la incertidumbre, estimar una cantidad adicional al coste que se ha identificado, generando lo que se conoce como “reserva de contingencia”. Se debe utilizar cuando a la actividad en la cual recae el coste le ha sido identificado algún riesgo. Para calcular esta reserva se utiliza el Análisis del Valor Monetario Esperado (VME), para lo cual, es necesario que el riesgo haya sido valorado de manera cuantitativa, es decir, que su impacto haya sido estimado en términos de dinero y/o tiempo. El Valor Monetario Esperado: es el resultado producto del impacto (consecuencia) del riesgo en dinero por su probabilidad de ocurrencia, esto es:

$$\text{VME} = \text{Impacto} \times \text{Probabilidad}$$

La Reserva de Contingencia será calculada a través de una combinación estadística de los Valores Monetarios Esperados. Para ello deberíamos combinar adecuadamente todos los sucesos, típicamente con herramientas de Simulación de Monte Carlo. Si

identificamos un riesgo que eventualmente pueda impactar positivamente al proyecto (Oportunidad) entonces el VME será un valor negativo y en consecuencia disminuye la reserva de contingencia.

En cuanto a la precisión de las estimaciones de coste, se debe tener en cuenta el momento en el que estas se llevan a cabo en el ciclo de vida del proyecto. En el PMBOK [69] se indica que al inicio del proyecto las estimaciones son de tipo ROM (*Rough Order of Magnitude*, Orden de Magnitud Aproximado o Bruto) con una precisión de -50% a +75%, cifra que se va reduciendo según las fases del proyecto se van desarrollando.

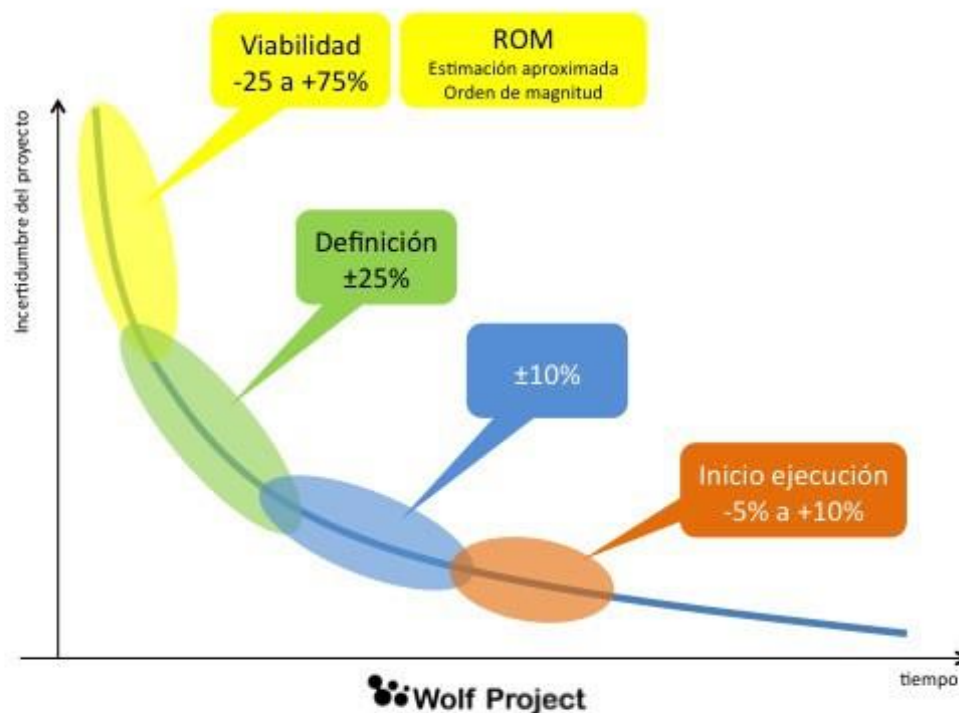


Figura 12 Estimación aproximada de orden de magnitud a lo largo de un proyecto [65].

Por otra parte, en el ámbito del Desarrollo de Software Ágil las técnicas de estimación tienen un gran peso debido al carácter iterativo y de entrega continua de los proyectos donde las estimaciones tienen que hacerse de forma progresiva. En [66] se realiza una clasificación general de los métodos de estimación de coste y esfuerzo:

- **Aproximamiento orientado al aprendizaje:** Basado en aprender de forma colectiva de las experiencias de estimación previas y el conocimiento de varios gestores obtenidos de los resultados de varios proyectos específicos. En el trabajo se critica a esta corriente porque suele proveer con estimaciones poco realistas.
- **Aproximamiento basado en expertos:** Un experto compara el proyecto con proyectos similares pasados basándose en la experiencia personal. El problema de este método es la alta subjetividad del mismo.

- **Métodos de regresión:** Este modelo se basa en datos de regresión y en desarrollar ecuaciones de regresión para hacer las estimaciones. Este método no promueve las buenas prácticas de la ingeniería de software.
- **Bottom-up:** Cada componente del sistema software se estima de forma separada y los resultados son agregados para producir una estimación del sistema completo. La dificultad reside en cómo descomponer el sistema en componentes estimables.

En [67] se realiza una revisión de las técnicas de estimación, que pueden ser incluidas en esa clasificación. Algunas de las más comunes son la opinión de expertos (similar al ya nombrado aproximamiento basado en expertos), Analogía y disgregación (aproximamiento basado en aprendizaje y regresión) y el *Planning Poker* (aproximamiento basado en expertos).

*Planning Poker* es una técnica de estimación basada en el consenso y la gamificación. Consiste en que los miembros del grupo de desarrolladores hacen estimaciones mostrando cartas numeradas, de forma simultánea, en vez de hacerlo de forma oral. Las puntuaciones reveladas son entonces discutidas y en caso de fuertes desacuerdos se procede a repetir la votación teniendo en cuenta la discusión. El uso de esta técnica evita el pensamiento en grupo ya que de forma oral, las primeras estimaciones sentarían un precedente para el resto [68]. En [67] se citan varios trabajos que concluyen que las estimaciones utilizando esta técnica mejoran a las realizadas de manera no estructurada o realizadas por expertos, aunque existe una dependencia de la experiencia de los practicantes de esta técnica.

### **3. Propuesta**

En esta sección se realiza la propuesta de un método para la elicitación y diseño de requisitos dirigido al valor del cliente incorporado en un proceso de desarrollo ágil de servicios en la nube y orientado al valor cliente.

En primer lugar se procederá a considerar el modelado del valor, explicando los objetivos de este, seleccionando un lenguaje para el modelado del valor y detallando las características de este. A continuación se expresa una vista general de proceso de desarrollo incorporando estas nociones de valor, detallando los elementos que lo forman, los entregables, roles y tareas del proceso. A continuación se detallarán los procesos más relevantes específicos de esta propuesta. En primer lugar el proceso propuesto de modelado del valor; en segundo las consideraciones relativas al proceso de priorización con elementos de valor y por último, estudiando la forma de unir los procesos de negocio con la representación del valor.

#### **3.1 El modelado de valor**

El modelado del valor es la parte vital del proceso en el cual se identificarán las consideraciones de valor del cliente y se traducirán a un modelo de valor. Para la descripción de los conceptos de valor se ha utilizado la aproximación de GRL, cuya descripción detallada se encuentra en sección 3.2.2. La intencionalidad de GRL y sus objetivos de modelado coinciden con la visión del uso del modelado de valor que se quiere aportar en este trabajo por lo que es una herramienta ideal para expresar el valor del cliente. Sin embargo, GRL no posee una metodología de modelado propia, es una herramienta de expresión por lo que en la presente sección se recoge una guía de actividades para facilitar el modelado, aportar directrices que permitan guiar el diseño de los modelos de manera que sean útiles en posteriores etapas del proceso y establecer claramente los objetivos de modelado.

##### **3.1.1 Objetivo**

El objetivo del modelado del valor es el de identificar el contexto en el que se va a incorporar el sistema. Esto significa que el modelado de valor debe incluir a todos los stakeholders cuyos objetivos de negocio (objetivos tangibles o intangibles, económicos o sociales, monetarios o utilitarios, estéticos o éticos) están relacionados con el sistema de información, así como la representación de estos objetivos y la relación entre ellos.

La representación de este contexto será la base para la justificación del sistema, el cual emergerá de las necesidades de cumplimiento de la organización, así como de las restricciones que el resto de stakeholders impongan sobre él.

Por todo ello la representación del valor debe:

- Representar y ayudar a comprender las motivaciones de la organización para el diseño del sistema de información y el dominio en el cual el sistema debe funcionar.
- Incluir los distintos stakeholders que rodean al sistema de información.
- Representar los objetivos de la organización, tanto de negocio, como relacionados con el sistema de información u de otra categoría relevante.

- Identificar las relaciones entre los objetivos de un stakeholder, el resto de objetivos propios y los de otros stakeholders o el sistema de información.
- Establecer un enlace de unión entre los objetivos a cumplir por el sistema de información y las consideraciones de valor del cliente y el resto de stakeholders.

### **3.1.2 Selección de un lenguaje de representación del valor**

Los requisitos del lenguaje seleccionado para representar el valor son:

- **Expresividad:** El lenguaje debe ser suficientemente expresivo como para poder representar cualquier tipo de elemento que represente valor para la organización, así como los distintos enlaces entre estos conceptos (contribuciones, dependencias, enlace con stakeholders, etc.). El valor puede adoptar muchas formas como se ha visto anteriormente (sección 2.4) por lo que el lenguaje debe ser capaz de plasmar todos estos conceptos que puedan suponer una restricción o un requisito para el sistema de información. Incluyendo, pero no limitándose a aspectos de negocio y flujo monetario.
- **Expresión de requisitos:** El lenguaje escogido debe permitir que la representación del valor pueda ser utilizada como una versión temprana del modelo de requisitos. A efectos prácticos este modelo debe ser un modelo CIM (*Computing Independent Model*) que capture el contexto del sistema y los requisitos de manera independiente de su implementación en un sistema de información concreto y con ello pueda ser utilizado en las distintas fases del proceso de desarrollo para el diseño o evolución de requisitos.
- **Extensibilidad:** El lenguaje escogido debe presentar mecanismos de extensión propios para modificar el lenguaje en el caso que fuese necesario por ejemplo para incluir nuevas propiedades de los elementos del modelo o crear nuevos tipos de elementos.
- **Madurez:** El lenguaje escogido debe contar con un nivel de madurez alto de manera que ya esté verificada su utilidad, exista una buena comunidad que soporte su uso y que incluya conocimiento, a ser posible, que facilite la creación de estos modelos, el aprendizaje del lenguaje así como las transformaciones entre distintos modelos de requisitos.

Es de relevancia que el lenguaje escogido forme parte de un estándar ya que esto garantiza un nivel de madurez, el soporte durante gran tiempo de este lenguaje así como una especificación única que facilita la interoperabilidad de las herramientas disponibles y de los modelos existentes.

El lenguaje escogido idealmente debe de poseer algún tipo de herramienta que soporte la creación y modificación de modelos de manera que el método pueda ser incorporado en un proceso de desarrollo sin necesidad del desarrollo de una herramienta específica para ello.

GRL es la solución que parece ajustarse más al lenguaje de modelado necesario para este proyecto. GRL se ajusta a la concepción de valor, más amplia vista en la sección 2.4 que engloba aspectos no solo económicos o estrictamente de negocio si no cualquier elemento de interés para los actores implicados. Estos conceptos son fácilmente formulables como aserciones en objetivos u objetivos soft ya que son elementos suficientemente amplios para incluir cualquier tipo de aspecto de los mencionados.

GRL a su vez permite asociar a cada objetivo u objetivo soft tareas que operacionalicen esos objetivos por lo que es posible realizar una primera aproximación al diseño del sistema, en un alto nivel de abstracción, de manera que la conexión entre requisitos y modelado de valor sea directa y pueda hacerse en la misma representación.

Con respecto a la extensibilidad, GRL en su versión estándar permite incorporar propiedades de cualquiera de sus elementos bajo el campo de metadatos, un mecanismo de extensión fácil y útil que permite realizar extensiones sencillas sin necesidad de extender el metamodelo de GRL, escapando del estándar; esto es también posible si fuera necesario ya que las bases de GRL están definidas con metamodelos y por lo tanto son extensibles usando las técnicas de la gestión de modelos.

GRL es un lenguaje maduro que cuenta con más de 20 años de antigüedad y ha sido estandarizado y revisado por última vez en 2009, con lo que cuenta con un importante aval. La ITU es una organización en la que se incluyen más de 190 países, por lo que es un estándar suficientemente respaldado. A su vez cuenta con una completa herramienta para la gestión de modelos GRL basada en eclipse, ampliamente extendido en la comunidad de desarrollo de software y especialmente en la de Desarrollo de Software dirigido por Modelos.

El mayor problema surge de su complejidad visual, que se incrementa al crecer el tamaño del modelo y resulta difícil de leer, a pesar de que GRL cuenta con pocos elementos y muy intuitivos. Este problema también está presente en el resto de lenguajes, pero puede mitigarse usando representaciones parciales (puntos de vista) u otro tipo de representaciones simplificadas para la comunicación con el cliente, puesto que lo más interesante del lenguaje es la ontología definida para la expresión de elementos de valor y no su representación gráfica.

i\* también ha sido firme candidato dada su similitud con GRL, el soporte de los conceptos de la orientación al valor vistos en i\* parecen contener la expresividad suficiente para utilizarla con el fin de representar el valor pero GRL, al no poseer una metodología propia, no está sometido a las restricciones del modelado de i\* o la intencionalidad de esta notación que no es estrictamente la misma que la del modelado del valor que se pretende conseguir en este trabajo.

A su vez, la falta de consenso entre las versiones de i\* frente a la estandarización de GRL hace que esta última sea mejor candidata.

Con respecto a e<sup>3</sup>-value, su mayor problema es de expresividad. A pesar de que ha demostrado ser útil para analizar flujos de valor monetario, es difícil expresar conceptos menos funcionales u obtener una representación del contexto de valor de la organización que sea completo y atienda a cualquier tipo de elemento de valor.

VDML es un estándar de nueva creación que apenas ha tenido tiempo para trabajarse, aunque prometedor en muchas de sus características y su visión holística de negocio y tecnologías, debería permitir obtener vistas de este tipo de modelos aptas para el personal de negocio y técnico. Sin embargo su orientación por el momento se acerca más a la visión de negocio y estratégica que a la de obtener requisitos operacionales más cercanos a la implementación. A su vez, su reciente aparición hace que escaseen herramientas específicas para la gestión de sus modelos, a pesar de ser un estándar MOF



y por tanto compatible con muchas de las herramientas que cumplen el estándar. De igual manera, el hecho de que combine muchas aproximaciones a la representación del valor (como la mencionada e<sup>3</sup>-value) hace que su expresividad sea ilimitada pero también hace que sea compleja de utilizar.

### **3.1.3 GRL en profundidad**

El principal objetivo del modelado del valor es explicitar los principales objetivos que contribuyen a la organización, cómo estos objetivos interactúan entre sí y qué tareas en los límites del sistema de información deben cumplirse y qué dependencias de servicios externos serán necesarias para cumplir los objetivos de valor. De manera que sea posible realizar la priorización de los requisitos a entregar utilizando el criterio de valor, identificando los elementos más valiosos (de mayor contribución al valor) y dividir el sistema para su entrega en iteraciones.

En un diagrama GRL de Valor se incluyen los objetivos, tareas, actores y sus relaciones que contribuyen de alguna manera al valor de la organización.

A continuación se describirán los distintos elementos del modelo:

#### **3.1.3.1 Actor**

Elemento que representa una entidad activa (un stakeholder, sistema u otro) que tiene intenciones y lleva a cabo acciones para conseguir sus objetivos ejerciendo su know-how. Un actor, posee un conjunto de valores que representan a su organización. Estos valores pueden descomponerse como un conjunto de objetivos cuyo cumplimiento representa el grado de eficacia de la organización. La frontera de un actor modela aquellos elementos intencionales que se encuentra, deben cumplirse o realizarse dentro de la organización. En el modelado el actor principal será el cual se esté focalizando ya que será el principal stakeholder de la solución para la cual se están elicitando los requisitos. El resto de actores externos a la organización añaden perspectiva al contexto del sistema y establecen restricciones sobre el actor sistema el cual modela las formas de satisfacción de los objetivos del resto de actores.

#### **3.1.3.2 Elementos intencionales**

Los elementos intencionales describen una intención. Su uso en el modelo del valor permite explicitar los elementos que pueden contribuir al valor de la organización (tanto de forma positiva como negativa). Son aquellos objetivos o tareas relevantes desde el punto de vista del valor de la organización, los actores externos y de la intencionalidad del sistema.

#### **3.1.3.3 Objetivos (goals)**

Un objetivo es una condición o estado del mundo que un actor querría conseguir. Cómo se consigue el objetivo no está especificado por el objetivo mismo. Un objetivo puede ser tanto un objetivo de negocio como uno del sistema o una aserción deseable para el actor, siempre que estos sean relevantes para el valor del actor. Un objetivo de negocio expresa objetivos de acuerdo con el estado de los asuntos de negocios que un individuo u organización querrían conseguir. Un objetivo de sistema expresa una meta que el sistema objetivo debe conseguir y generalmente describe objetivos funcionales del sistema de información objetivo (cuyos requisitos queremos capturar). Un objetivo conseguido contribuye de forma positiva al valor de su actor mientras que uno no conseguido puede reducir el valor del mismo.

### **3.1.3.4 Objetivos soft (soft-goals)**

Un objetivo soft es una condición o estado del mundo que un actor querría conseguir pero que, a diferencia del concepto de objetivo, no existe un criterio que determine de forma exacta si la aserción del objetivo se ha cumplido y es responsabilidad de un juicio subjetivo y la interpretación del modelador juzgar si un particular estado de las cosas en efecto consigue suficientemente el objetivo soft definido. Este concepto se suele aplicar a cualidades y aspectos no funcionales como la seguridad, la robustez, el rendimiento o la usabilidad (entre muchos otros) que poseen una escala cuantitativa de cumplimiento a la que es necesario establecer límites para valorar su cumplimiento o no. De igual manera, el cumplimiento de un objetivo soft incrementa el valor de un actor.

### **3.1.3.5 Tareas (Tasks)**

Una tarea específica una forma particular de hacer algo. Cuando una tarea es parte de una descomposición de otra de mayor nivel, restringe esa tarea de nivel más alto a un curso de acción particular. Las tareas representan las formas de cumplimiento de los objetivos. Una tarea debe contribuir con al menos un objetivo (hard o soft). Una tarea está relacionada en un mayor nivel de abstracción, con una actividad o conjunto de actividades de proceso de negocio que estarán soportadas por el sistema de información.

### **3.1.3.6 Enlaces**

Un enlace conecta dos elementos intencionales. Existen tres tipos de enlaces de dependencia, contribución o descomposición:

#### **3.1.3.6.1 Dependencia**

Una dependencia conecta dos actores a un elemento intencional. Describe cómo un actor depende en otro para cumplir el elemento intencional que los une. En el caso del dominio de software en la nube esta relación puede modelar el consumo de servicios de un tercero.

#### **3.1.3.6.2 Contribución**

Un enlace de contribución une un elemento intencional con otro sobre el que tiene un efecto (positivo o negativo). Un enlace de contribución puede unir objetivos, objetivos flexibles, y tareas. El uso de los enlaces de contribución debe tratar de representar cualquier efecto que los elementos intencionales tengan entre sí que pueda condicionar la selección de requisitos del sistema o que afecten de alguna manera al valor del cliente. Estos efectos tendrán un papel decisivo a la hora de priorizar requisitos (ver sección 3.3).

#### **3.1.3.6.3 Descomposición**

En ocasiones un elemento intencional (tareas, objetivos y objetivos flexibles) puede ser descompuesto en unidades más pequeñas, para detallar en más profundidad un elemento relevante para el cliente.

Las descomposiciones pueden ser del tipo AND: descomposición jerárquica del elemento intencional objetivo por un elemento fuente en la que la satisfacción de cada uno de los elementos sub-intencionales es necesaria para conseguir el objetivo; y del tipo OR la cual describe las alternativas de para satisfacer un elemento intencional objetivo.

## **3.2 Vista general del proceso de desarrollo dirigido por el valor**

El proceso que se va a presentar a continuación tiene por objetivo soportar el desarrollo incremental de software orientado al valor del cliente, este proceso puede incluirse dentro del proceso general de desarrollo de un proyecto ágil y solo se especifican las

actividades y elementos relevantes para llevar a cabo las partes relacionadas con la especificación del valor y su uso para la especificación de requisitos de producto software y cómo priorizar esta lista de requisitos para ser incluidos en *backlog* para su implementación en una primera iteración, ya sea un sprint de un proyecto ágil que utilice sprint u otro concepto similar de otra metodología de desarrollo.

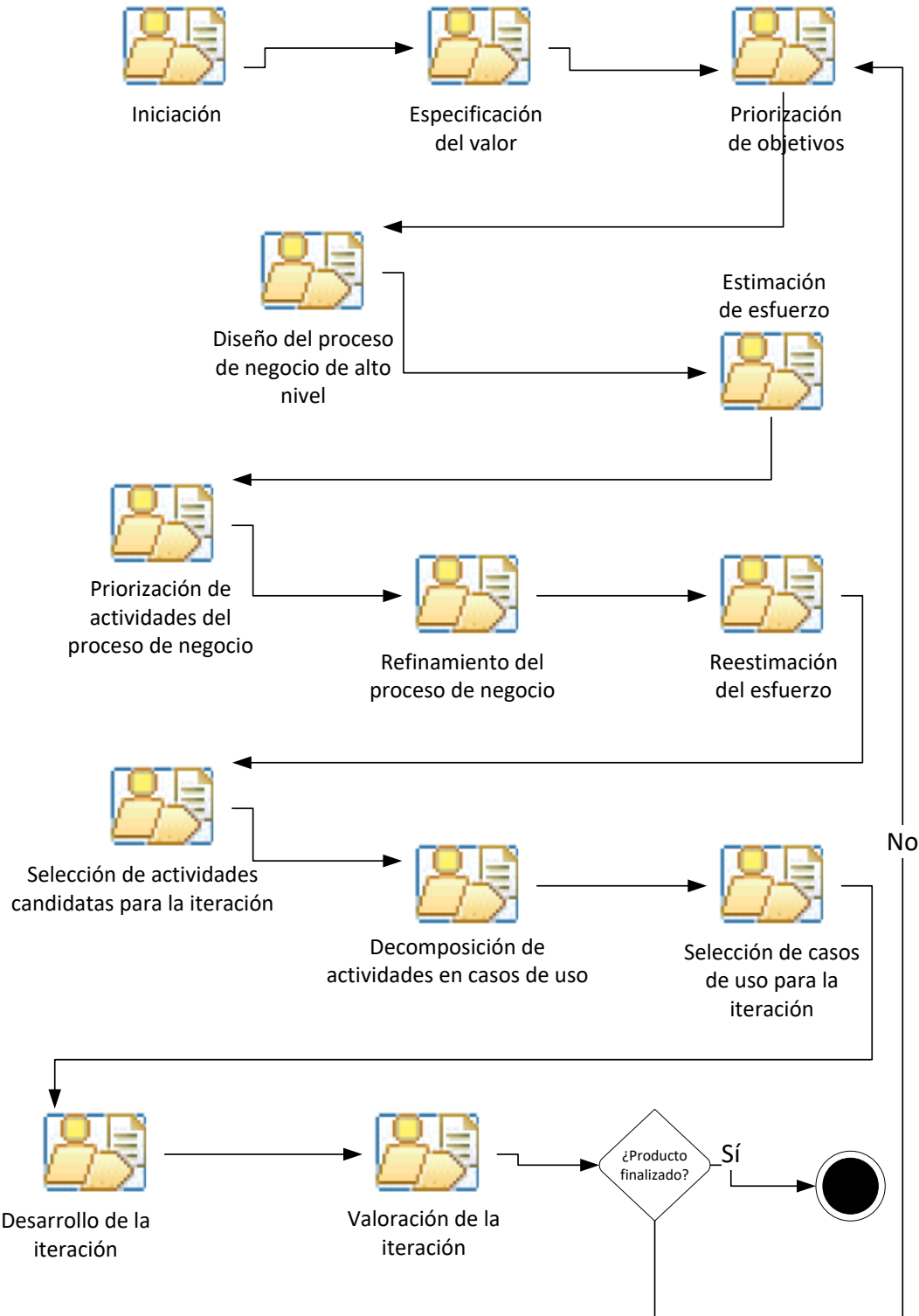


Figura 13 Diagrama de actividad en el que se enumeran las distintas actividades del método y su orden secuencial.

Este proceso combina las prácticas ágiles con la ingeniería de modelos para la modelización del valor y su incorporación al proceso de captura de requisitos. Este proceso emplea como lenguaje para la descripción del valor GRL, como descriptor de los procesos de negocio BPMN y la descripción de requisitos software. A continuación se procede a exponer la vista general del proceso y a explicar cada una de sus fases. En las secciones posteriores se detallarán cada una de ellas así como, si fuera necesario, las técnicas o herramientas utilizadas.

### 3.2.1 Iniciación

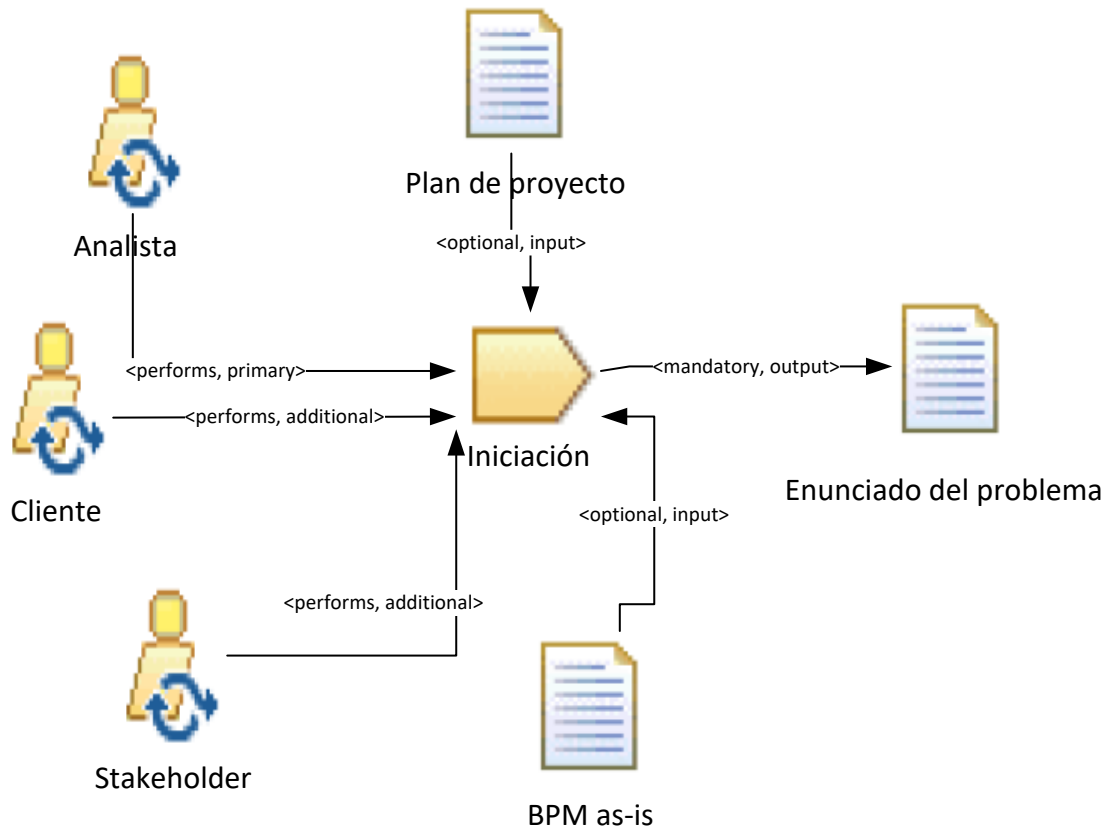


Figura 14 Iniciación

#### Descripción

Esta es la primera actividad del proceso. Una vez el proyecto ha sido aprobado por dirección se procede a iniciar la fase de elicitación de requisitos. Como preparación a esta fase es necesario acordar primero los términos del proyecto, familiarizarse con la organización y la necesidad a cubrir por el proyecto software así como fijar las restricciones temporales del proyecto.

En esta fase se toma como referencia, si existe, la solicitud del proyecto, ya que aporta parte de esta información y se comienza a analizar la organización tomando como partida su proceso de negocio actual (si existe) y deberá ser especificado en BPMN para tomarse en cuenta en el desarrollo del sistema.

Se identificarán los distintos procesos a los que debe dar soporte la aplicación y los analistas comenzarán a analizar el proceso de negocio para tratar de comprender la solución necesaria.

### Roles

- Representante de dirección de la empresa cliente.
- Otros stakeholders del contexto del sistema.
- Analista.

### Artefactos

- Proceso de negocio *as-is*.
- Plan de proyecto.
- Enunciado del problema.
- Artefactos del proceso de negocio a dar soporte.

### 3.2.2 Especificación del valor

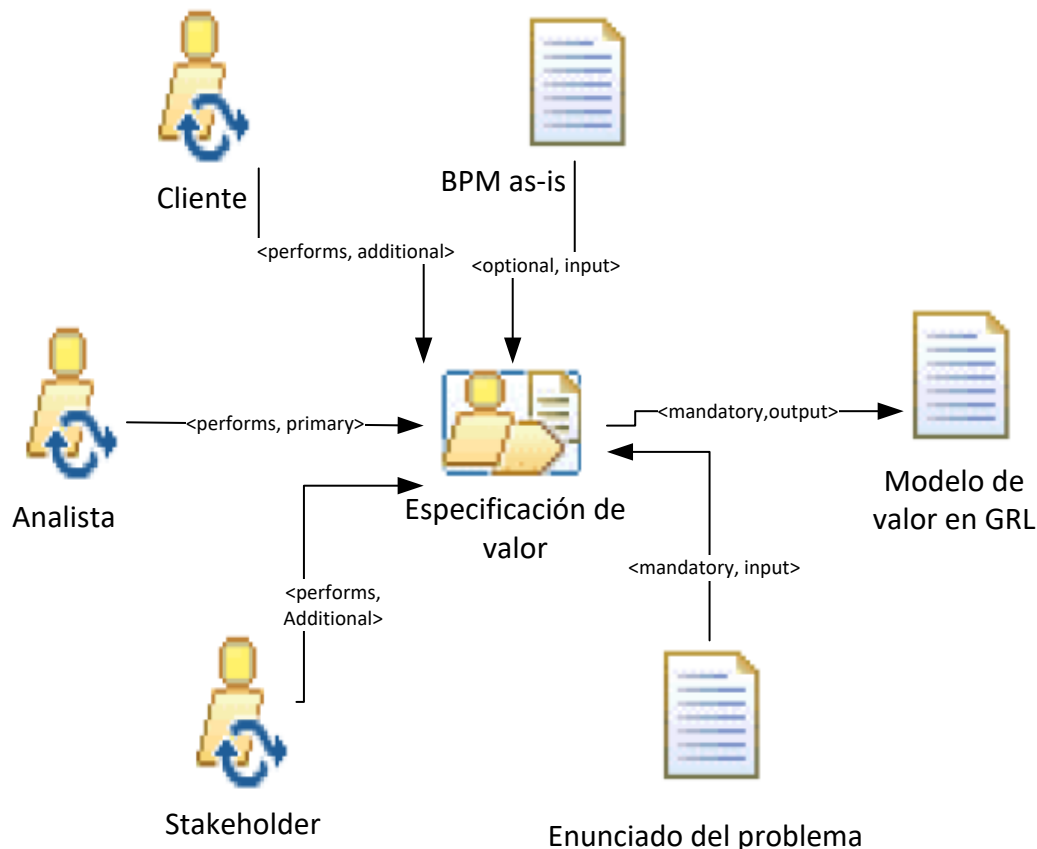


Figura 15 Especificación del valor

### Descripción

Este subproceso comprende un conjunto de actividades cuyo objetivo es el de establecer un modelo de valor que sirva a dos objetivos: modelar los elementos de valor del cliente y el resto de stakeholders, es decir representar los conceptos de interés para cada una de las partes que el sistema debe de tener en cuenta; establecer los requisitos de alto nivel que deberá tener el sistema y establecer una relación entre ambos. Este modelo debe

representar el porqué de la funcionalidad presente en el sistema y se utilizará como entrada para la priorización de objetivos.

Este proceso se encuentra detallado en la sección 3.3.

### Roles

- Analista
- Cliente.
- Representante/s de los stakeholders.

### Artefactos

- Modelo de valor en GRL.
- Modelo de proceso de negocio *as-is*.
- Enunciado del problema.

### 3.2.3 Priorización de Objetivos

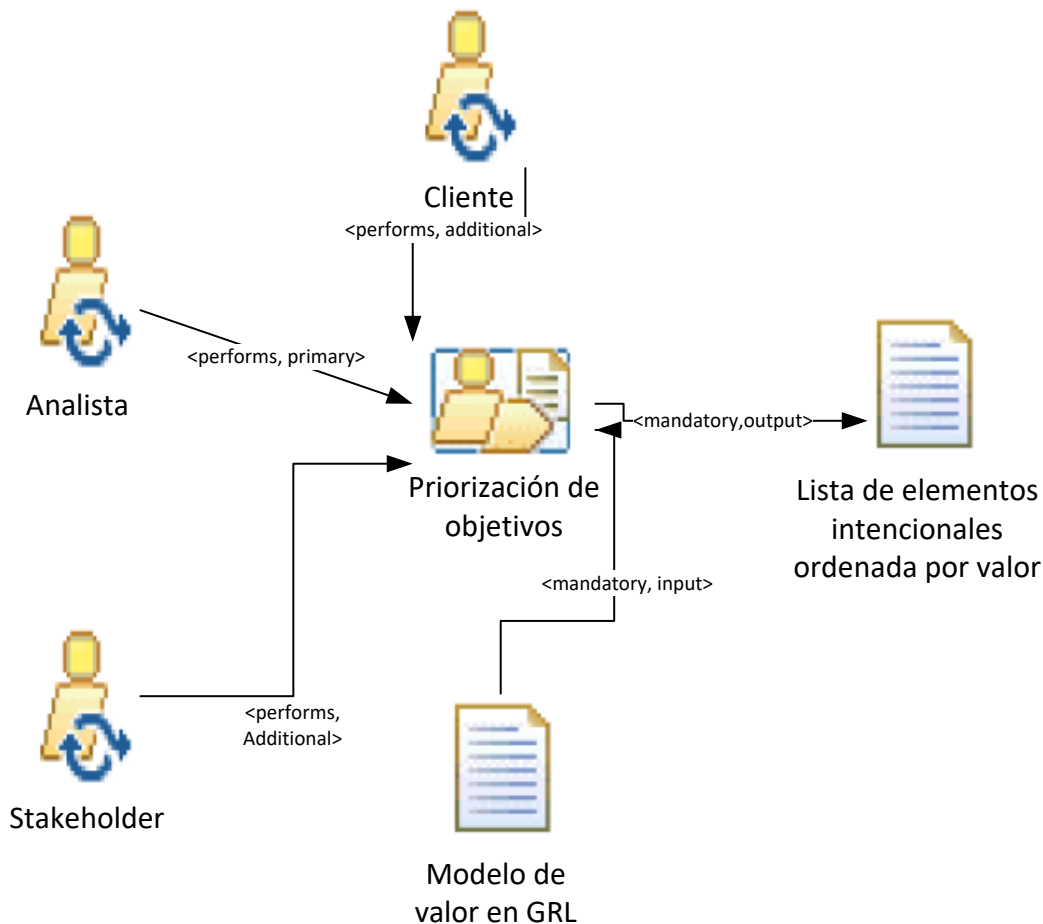


Figura 16 Priorización de objetivos.

### Descripción

Tomando como entrada el modelo de valor en GRL, se deberá proceder a priorizar los objetivos que lo conforman, de manera que esta priorización pueda ser extendida a los requisitos del sistema. Esta actividad se encuentra explicada con mayor detalle en 3.4.2,

utiliza *Value Point Analysis*, propiedades de los elementos de GRL y un algoritmo de propagación de la importancia para establecer una lista ordenada según importancia de objetivos.

### Roles

- Analista.
- Representante/s de los stakeholders.
- Cliente.

### Artefactos

- Modelo de valor en GRL.
- Lista de elementos intencionales ordenada por valor.

## 3.2.4 Diseño del proceso de negocio de alto nivel de abstracción

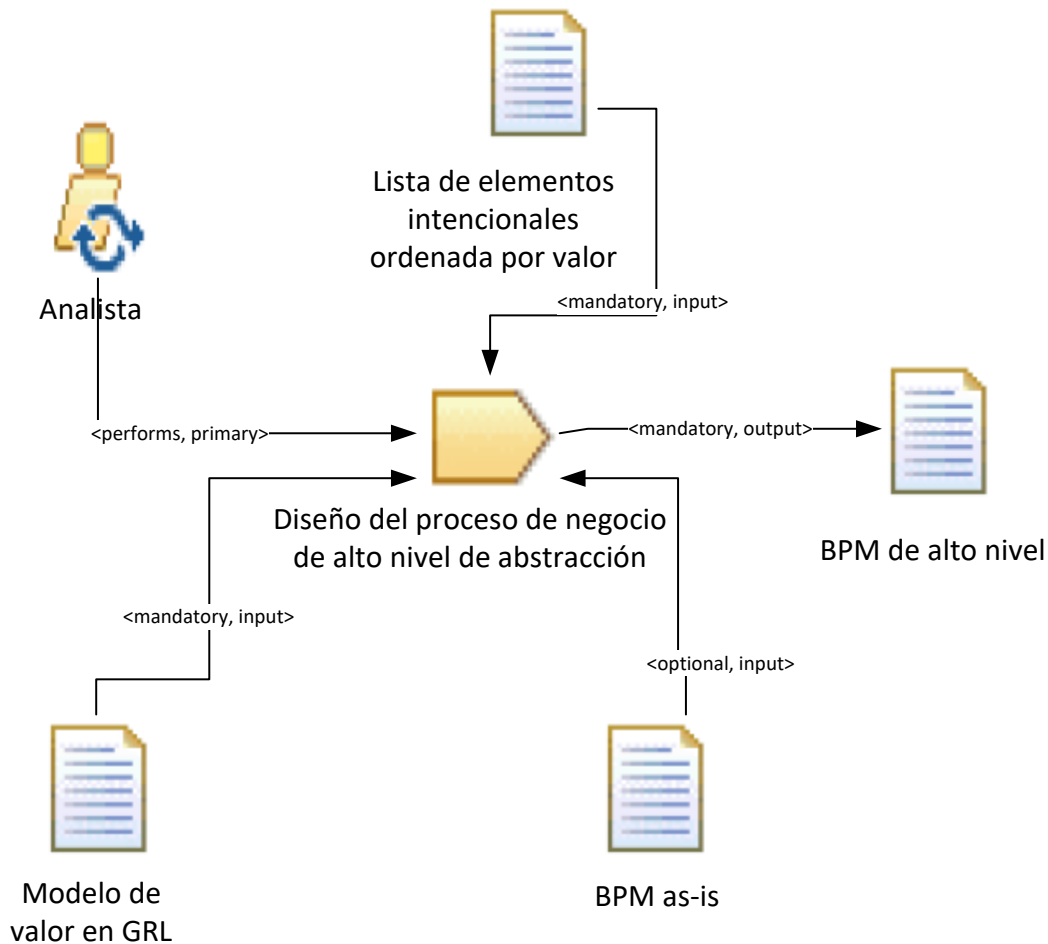


Figura 17 Diseño del proceso de negocio de alto nivel de abstracción.

### Descripción

En esta actividad, tomando como punto de partida la lista de objetivos priorizada y el modelo de valor, se comienza a desarrollar el modelo de proceso de negocio en BPMN, utilizando técnicas de transformación de modelos y la información del mismo para construir un modelo de proceso de negocio de baja granularidad, en el que se muestren

los principales procesos y los roles que los desempeñan. Posteriormente este proceso de negocio se detallará hasta llegar a las unidades de funcionalidad básicas. Como resultado se obtendrá el modelo de proceso de negocio to-be de alto nivel de abstracción.

### Roles

- Analista

### Artefactos

- Lista de elementos intencionales ordenada por valor.
- Modelo de proceso de negocio de alto nivel de abstracción (to-be).
- Modelo de valor en GRL.
- Modelo de proceso de negocio *as-is*.

## 3.2.5 Estimación de esfuerzo

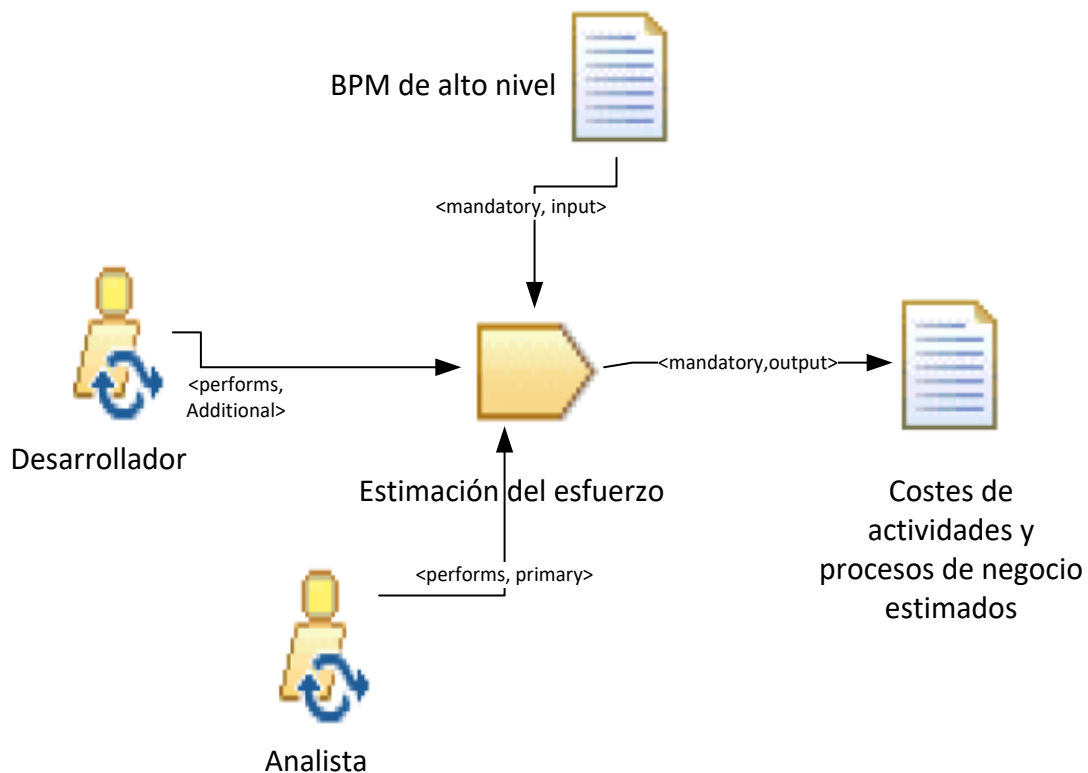


Figura 18 Estimación del esfuerzo.

### Descripción

En esta actividad se estimará el esfuerzo de implementación con alto nivel de granularidad de cada uno de los subprocessos del modelo de negocio de alto nivel de abstracción establecido en el paso anterior. Esta es una estimación todavía preliminar de los procesos de negocio del modelo de proceso a alto nivel. Por ello esta estimación se realizará de manera relativa entre actividades, su propósito es identificar qué procesos será más costosos que el resto no el coste exacto de desarrollo de estos procesos. Más detalles sobre la estimación de costes se encuentran en la sección 3.4.2. Esta estimación necesita de un nivel de conocimiento previo por parte de expertos en el dominio del software a desarrollar o de datos históricos de otros desarrollos similares de manera que



los diferentes procesos puedan ser comparados con datos reales y acordes al equipo de desarrollo de la organización.

### Roles

- Analista.
- Desarrollador.

### Artefactos

- Modelo de proceso de negocio de alto nivel de abstracción.
- Listado de costes de actividades y procesos de negocio.

### 3.2.6 Priorización de actividades del proceso de negocio

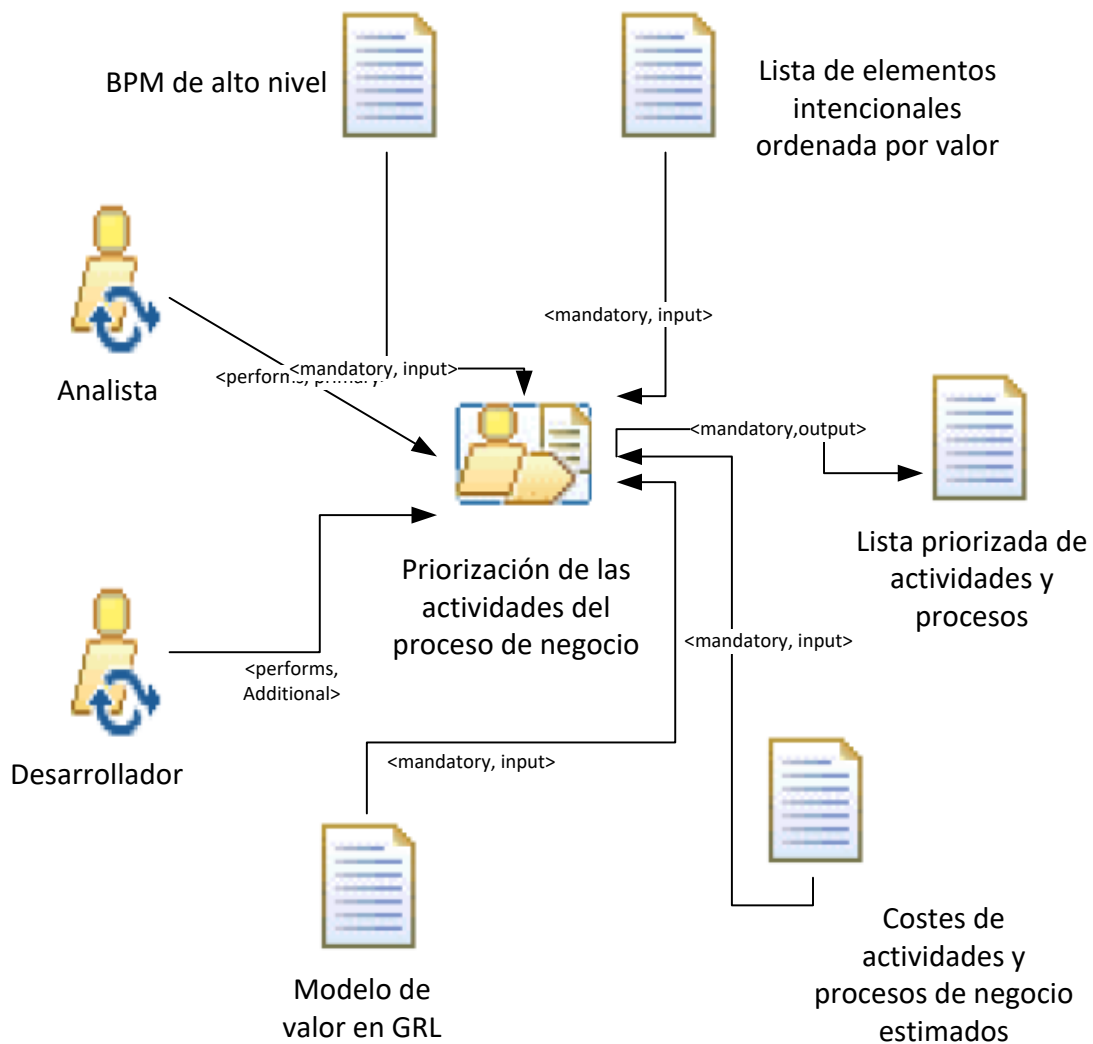


Figura 19 Priorización de actividades del proceso de negocio.

### Descripción

En esta actividad se procederá a utilizar el coste estimado y el valor aportado estimado para realizar una nueva priorización de las actividades de alto de nivel elicidadas en las tareas anteriores. De esta manera se observarán los dos componentes de principal

prioridad para la selección de actividades del sprint. Para el método detallado de creación de esta lista de elementos prioritarios consultar la sección 3.4.2 de este documento.

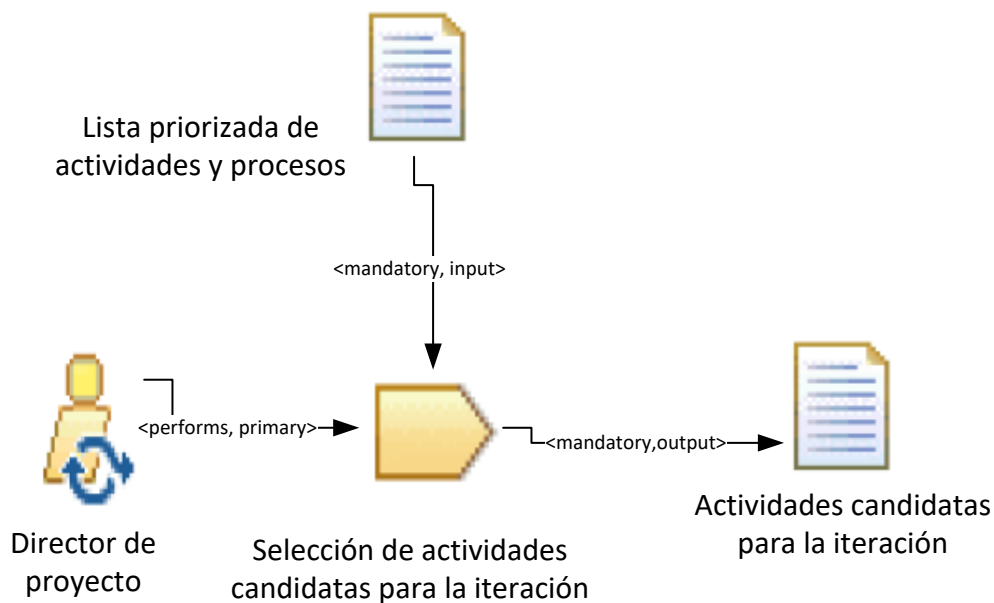
### **Roles**

- Analista.
- Desarrollador.

### **Artefactos**

- Lista de actividades de negocio priorizada por coste y valor.
- Modelo de proceso de negocio de alto nivel de abstracción.
- Actividades y procesos con costes asignados.

## **3.2.7 Selección de actividades candidatas para la iteración**



**Figura 20 Selección de actividades candidatas para la iteración.**

### **Descripción**

Tomando el listado de actividades de negocio priorizada por coste y valor se seleccionan las actividades que a priori, con los datos de coste estimado anteriormente, formarán parte de la iteración de desarrollo. Se seleccionarán las actividades que completen (o más se acerquen) a la capacidad de trabajo del equipo de desarrollo (para su cálculo se puede acudir a diferentes prácticas, utilización de datos históricos del equipo de desarrollo, utilización de datos promedios, etc.) de manera que se maximice el valor entregado por esa iteración en forma de sprint.

### **Roles**

- Director de proyecto.

### **Artefactos**

- Lista de actividades de negocio priorizada por coste y valor.

- Listado de actividades del proceso de negocio candidatas para la iteración.

### 3.2.8 Refinamiento de procesos de negocio candidatos

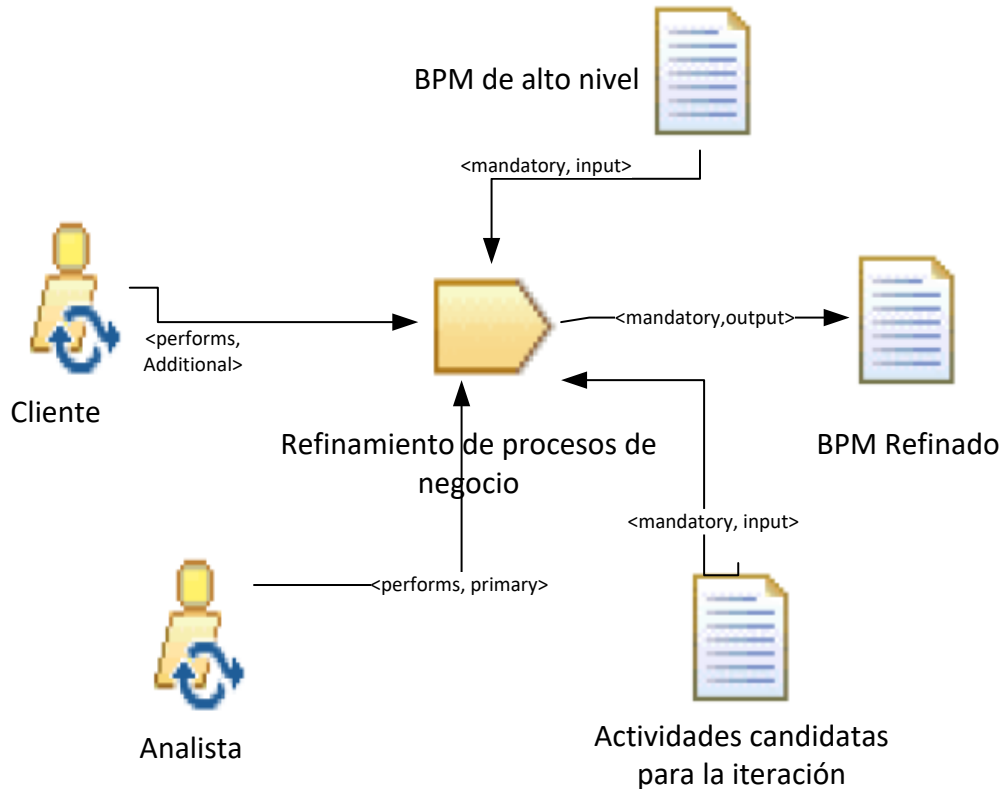


Figura 21 Refinamiento de procesos de negocio.

#### Descripción

Una vez seleccionadas las actividades candidatas para la iteración se procede a detallar el diseño de estos procesos de negocio, incluyendo más detalles del flujo de trabajo, expandiendo aquellos procesos que lo estuvieran y detallando actividades demasiado generales. Esta actividad del proceso tiene un triple objetivo: refinar los procesos de alto nivel elicitados, acercar al nivel de casos de uso trasladables a unidades de trabajo para que los desarrolladores puedan trabajar en ellas en la interacción posterior y la reestimación del coste de las mismas, una vez se cuente con más información que permita corregir las estimaciones iniciales. La intervención en este proceso por parte del cliente al menos a nivel de validación es deseable.

#### Roles

- Analista.
- Cliente.

#### Artefactos

- Listado de actividades del proceso de negocio candidatas para la iteración.
- Modelo de proceso de negocio de alto nivel de abstracción.
- Modelo de proceso de negocio refinado.

### 3.2.9 Reestimación de esfuerzo

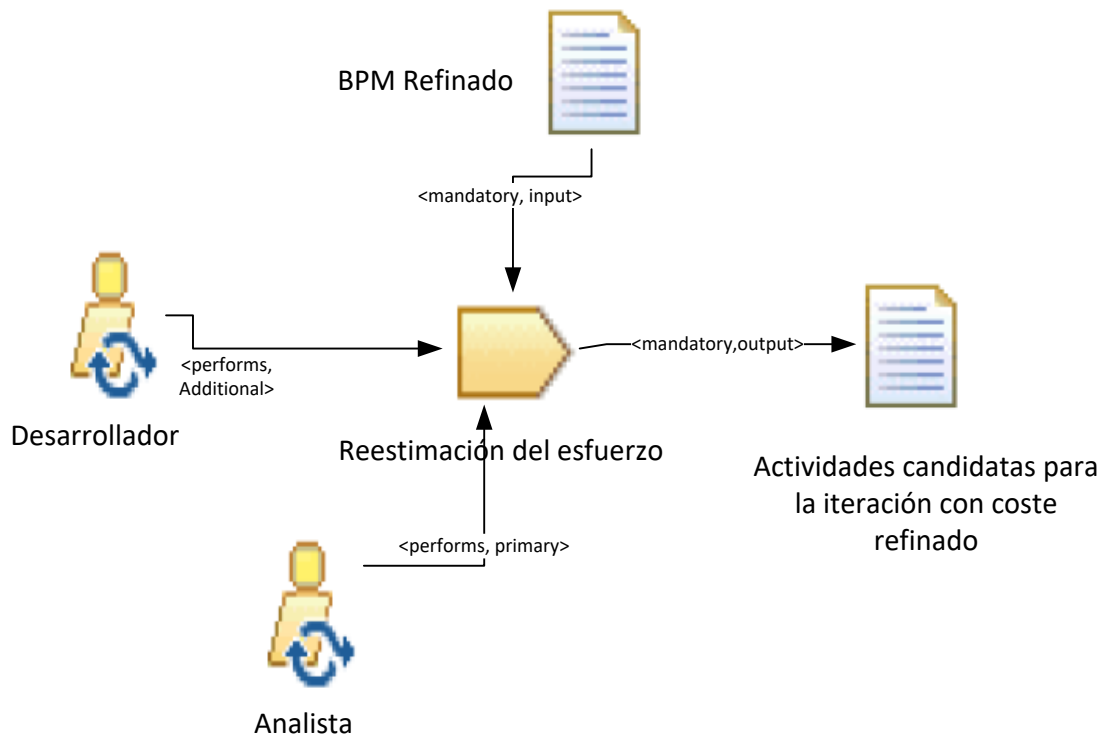


Figura 22 Reestimación del esfuerzo.

#### Descripción

Dado que la primera estimación del esfuerzo de desarrollo de los procesos de negocio se ha realizado con un alto nivel de abstracción, es necesario utilizar la nueva información aportada por el refinamiento de los procesos de negocio para obtener una expresión cuantificada de los costes de los nuevos procesos para que se reduzca el margen de error en la estimación y sea posible utilizar el criterio de coste para su inclusión en un sprint.

#### Roles

- Analista.
- Desarrollador.

#### Artefactos

- Modelo de proceso de negocio refinado.
- Listado de actividades del proceso de negocio candidatas para el sprint reestimadas.

### 3.2.10 Reselección de actividades candidatas para la iteración

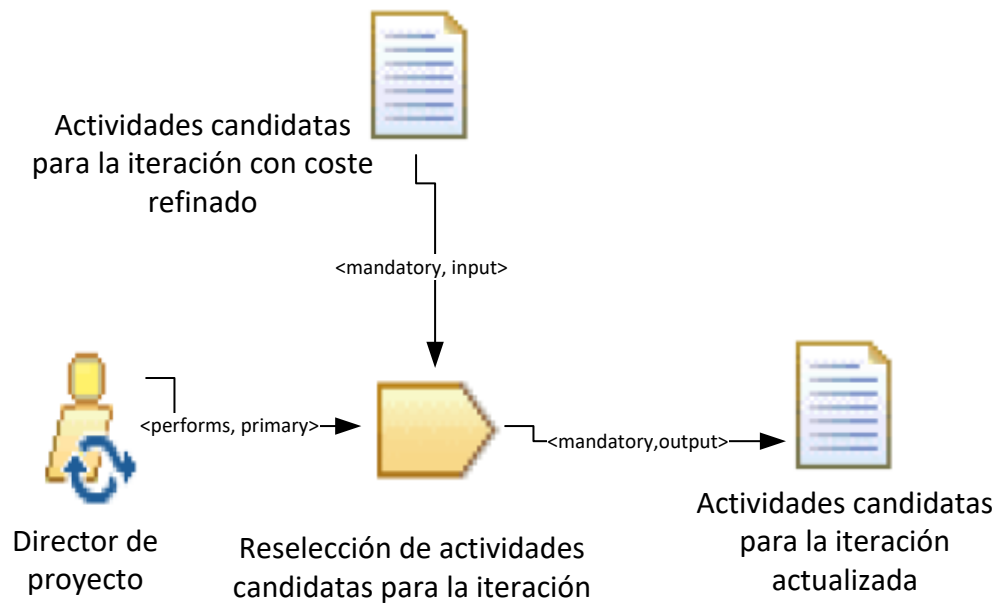


Figura 23 Reselección de actividades candidatas para la iteración.

#### Descripción

En esta actividad se procede de nuevo a la selección (si fuera necesaria) de aquellas actividades que una vez reestimadas maximicen el valor entregado al cliente y minimicen los costes.

#### Roles

- Director de proyecto.

#### Artefactos

- Listado de actividades del proceso de negocio candidatas para la iteración reestimadas.
- Listado de actividades del proceso de negocio candidatas para la iteración reestimadas y reseleccionadas.

### 3.2.11 Descomposición de actividades en casos de uso

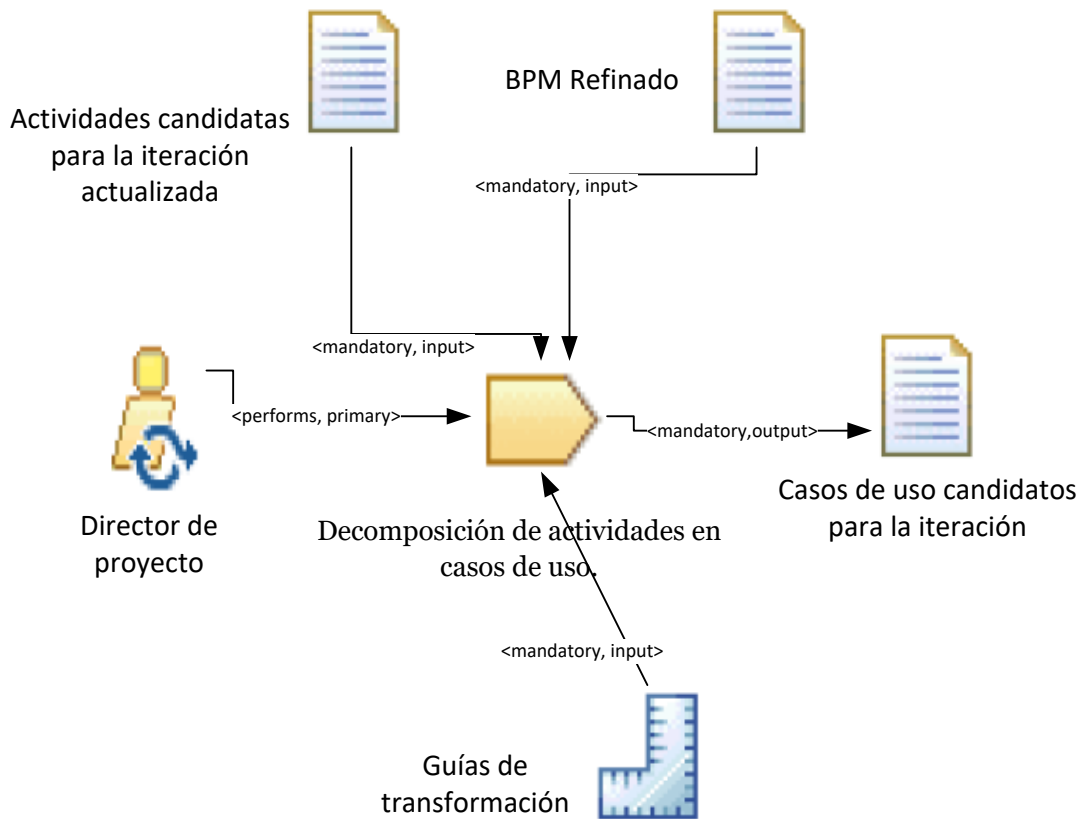


Figura 24 Descomposición de actividades en casos de uso.

#### Descripción

En esta actividad las actividades seleccionadas para el sprint son transformadas en casos de uso que serán los elementos con los que finalmente trabajarán los desarrolladores durante el desarrollo de las iteraciones.

#### Roles

- Analista.

#### Artefactos

- Listado de actividades del proceso de negocio candidatas para el sprint reestimadas y reselectionadas.
- Listado de casos de uso candidatos para el sprint.
- Guías de transformación de actividades de BPMN en casos de uso UML.

### 3.2.12 Selección de casos de uso para la iteración (preparación del sprint)

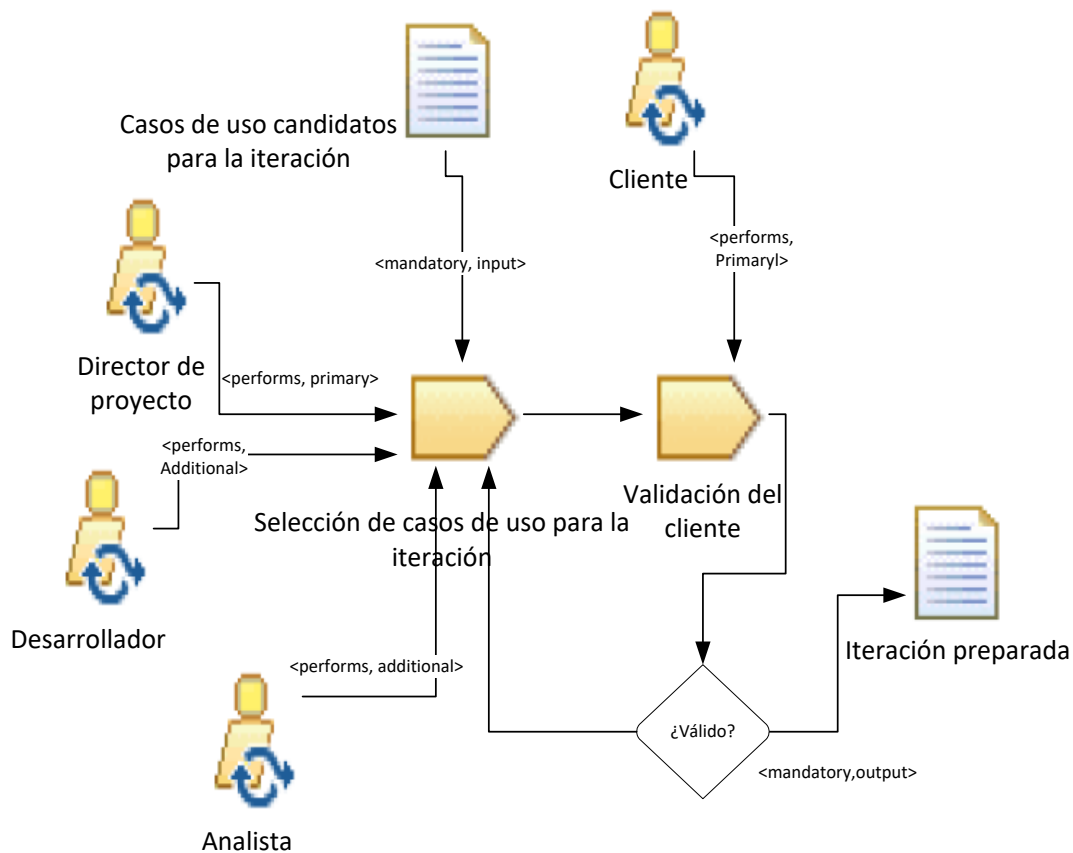


Figura 25 Selección de casos de uso para la iteración.

#### Descripción

En esta actividad se realiza el símil a una preparación del sprint tradicional en los métodos ágiles. Esto es, se seleccionan aquellos casos de uso que formarán parte de la lista que se implementará durante el sprint de manera que el coste (esfuerzo) total de los casos de uso seleccionados no supere la capacidad de trabajo del equipo de desarrollo y garantizando que estos casos de uso son los que contribuyen en mayor cuantía al valor del cliente ya que proceden de un proceso de análisis y refinamiento del valor.

Si el coste de los casos de uso excediera la capacidad del trabajo para el sprint se procedería a reservar esos casos de uso para el sprint posterior, teniendo estos prioridad sobre los nuevos casos de uso.

La propuesta del sprint debe de ser validada con el cliente, pudiendo este alterar el orden de la priorización si las circunstancias lo requieren.

#### Roles

- Analista.
- Desarrollador.
- Cliente.
- Director Proyecto.

## Artefactos

- Listado de casos de uso candidatos para el sprint.
- Iteración preparada.

### 3.2.13 Desarrollo de la iteración

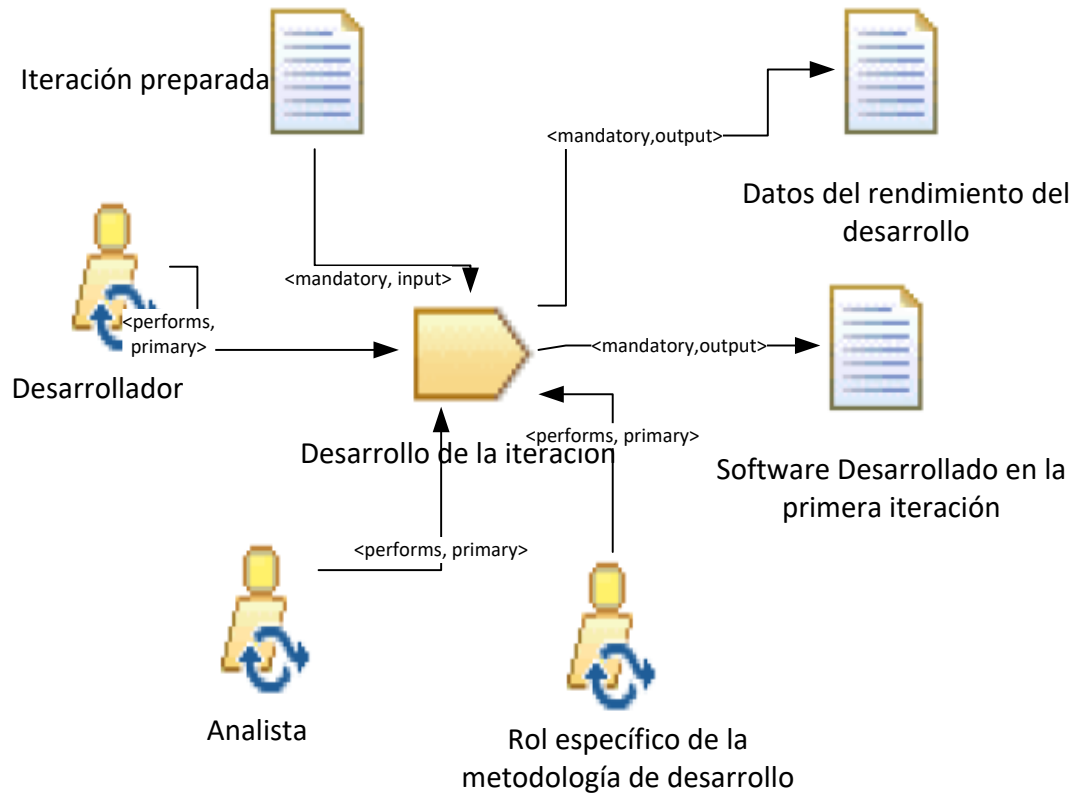


Figura 26 Desarrollo de la iteración.

## Descripción

En este proceso se encuentran las actividades convencionales del proceso de desarrollo de cualquier metodología ágil como SCRUM que emplearía como backlog del sprint el creado por la actividad anterior, es decir la iteración preparada en base a los casos de uso derivados.

## Roles

- Analista
- Desarrollador.
- Otros roles propios de la metodología ágil empleada (*Scrum Master, Product Manager...*)

## Artefactos

- Iteración preparada.
- Software desarrollado en la iteración.
- Datos del rendimiento del desarrollo.



### 3.2.14 Análisis de la iteración

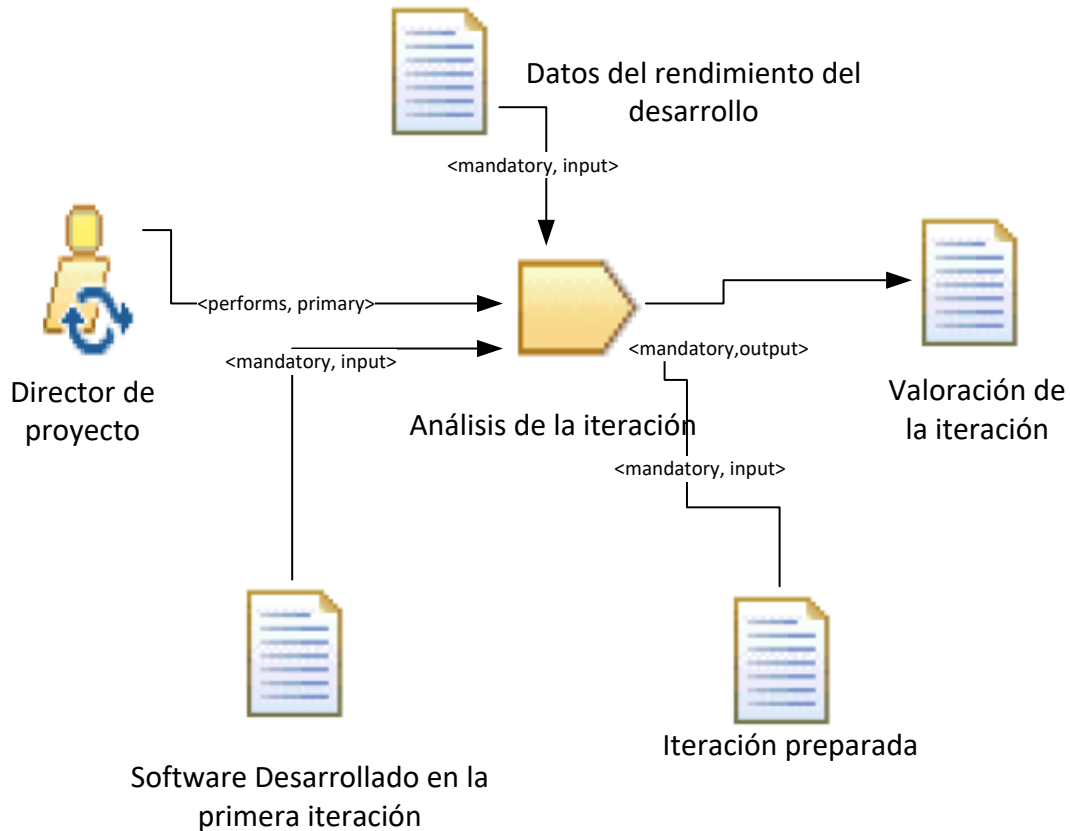


Figura 27 Análisis de la iteración.

#### Descripción

Se procede al análisis de los resultados de la iteración analizando los datos aportados por el proceso comprobando la calidad de la estimación de costes, las tareas no terminadas si las hubiese, errores y aciertos cometido durante el desarrollo, etc. que servirán para la retroalimentación del proceso.

#### Roles

- Director de proyecto.

#### Artefactos

- Iteración preparada.
- Software desarrollado en la iteración.
- Datos del rendimiento del desarrollo.
- Valoración de la iteración.

### 3.3 El proceso de modelado del valor

El proceso de modelado del valor es una técnica de la ingeniería de requisitos a utilizar en una fase temprana de la elicitación de requisitos. Esta técnica se emplea como se ha explicado en el apartado anterior para conocer de manera más profunda el contexto del sistema de información cuando apenas se tiene información de este y de los requisitos que deberá contener. El proceso tiene por objetivo la elaboración de un modelo de valor

que está clasificado en la arquitectura de modelos como Modelo Independiente de la Computación (CIM, *Computing Independent Model*), estos modelos se encuentran en un nivel de abstracción superior que sirven de puente entre los expertos del negocio y los desarrolladores que afrontan la realización del sistema. Los modelos de este nivel no representan detalles del sistema en sí, sino del dominio de aplicación del mismo.

El proceso propuesto a continuación se compone de una serie de actividades que deben realizarse para conseguir un modelo completo. El orden establecido es el conceptualmente más adecuado para construir el modelo de forma lineal, sin embargo el proceso de elicitación de requisitos es iterativo y flexible, dado que la exploración de la solución implica el descubrimiento de nuevos puntos de vista o la clarificación de la información implica la vuelta sobre las diferentes actividades.

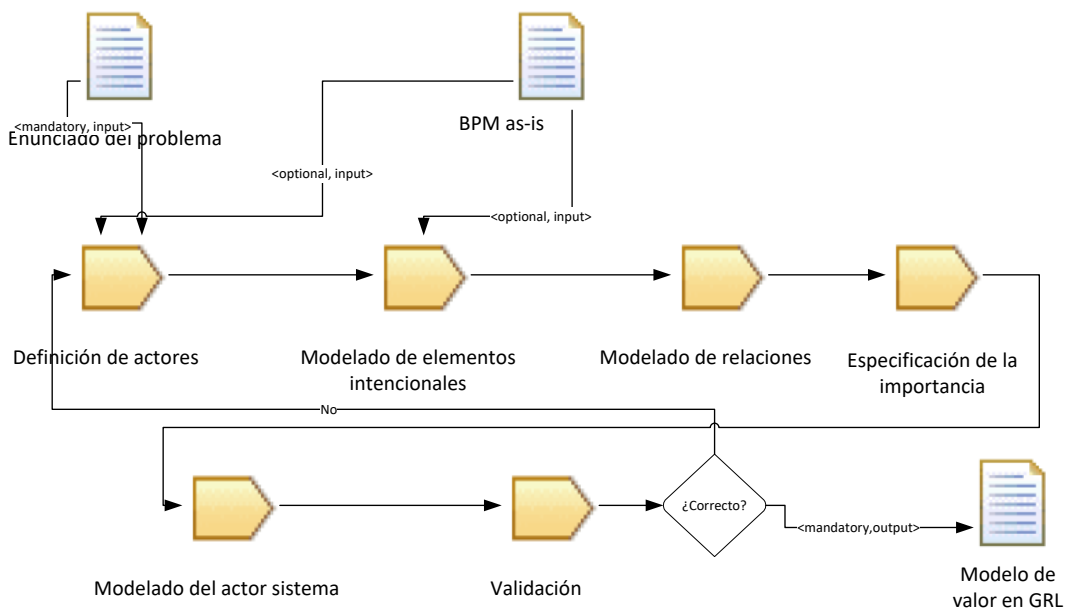


Figura 28 Vista general del proceso de modelado de valor

### 3.3.1 Definición de actores

En esta actividad se especifican cuáles son los stakeholders que rodean al sistema de información. Estos stakeholders serán modelados en GRL utilizando el elemento Actor. De ahora en adelante este será el término para referirse a los stakeholders dentro del modelo de valor.

Según el papel que los actores desempeñen en el contexto de la organización estos pueden clasificarse en:

- **Actor principal:** Organización en la cual se va a implementar el SI y que por tanto es el foco del que se obtendrán los requisitos y los objetivos que se tratarán de satisfacer con la implementación del SI.
- **Actor Sistema:** Representación del sistema de información como contenedor de los requisitos y objetivos que debe cumplir la solución a desarrollar.
- **Actor Sistema legado:** Si fuera pertinente, la existencia de un sistema legado que debiera tenerse en cuenta en el modelado del valor, deberá representarse por separado del sistema nuevo.

- **Actores externos:** Stakeholders relevantes del entorno de la organización: Clientes, proveedores, organizaciones aliadas, organismos gubernamentales... Forman parte del contexto del negocio e imponen restricciones al sistema de información u objetivos de la organización.

### 3.3.2 Modelado de los actores

En esta actividad el actor principal y los actores externos identificados en la anterior son expandidos representando los objetivos que constituyen el comportamiento del actor, esto se expresa mediante los elementos intencionales de GRL elementos intencionales objetivos, objetivos soft y tareas. El aproximamiento recomendado para definir estos elementos es *top-down*, especificando primero los conceptos más generales y refinando, mediante enlaces de descomposición OR o AND, aquellos que así lo requieran. Debe recordarse que este es un modelo de tipo CIM así que no deben incluirse detalles relativos a la cómo los elementos aparecerán en el apartado técnico de la solución

El modelado de los actores es ligeramente diferente en su aproximación a los distintos tipos de actor, aunque existen unas directrices generales a la hora de realizar el modelo:

- **Modelado de elementos intencionales (objetivos, objetivos soft, tareas):** Son el aspecto más relevante del actor. Cada elemento debe representar un objetivo, tarea, necesidad o restricción relevante para el actor y de un nivel de abstracción elevado. Estos elementos se encuentran en el interior de las fronteras del actor lo que significa que el actor es el afectado por el cumplimiento o no del elemento intencional.
- **Modelado de relaciones:** En esta fase se modelan las relaciones entre los elementos intencionales dentro de las fronteras del actor. Estas relaciones pueden ser de dos tipos composición o contribución:
  - Composición (AND, OR): Se modelan la descomposición en tareas mediante enlaces and o las alternativas mediante enlaces OR.
  - Contribución: Se modelan las relaciones entre elementos intencionales indicando la forma en la que afectan al resto de elementos de forma positiva o negativa, en una escala finita cuantitativa (consultar detalles en el apartado 3.1.3, GRL en profundidad).
- **Especificación de la importancia:** Establecer la importancia para el actor el cumplimiento de cada elemento intencional en una escala finita cuantitativa que va de 0 a 100 (consultar más detalles en la sección 3.3 de priorización de requisitos).

Los diferentes actores presentan algunas particularidades de perspectiva a la hora de modelar:

- **Actor principal:** es el principal foco del modelado de valor, de los elementos modelados en este actor dependerán en gran medida los objetivos del sistema y por tanto los requisitos derivados. Es vital incluir todos los objetivos relevantes para la organización y definir claramente las relaciones de contribución entre los distintos elementos.
- **Actores externos:** El modelado de los stakeholders no debe hacerse con un detalle similar al del actor principal, deben incluirse solamente

aquellos elementos que tengan impacto en el actor principal o en el sistema ya que estos supondrán restricciones al sistema.

### **3.3.3 Modelado del actor sistema**

Mientras que el actor principal y los actores externos poseen propiedades extraíbles del dominio a priori, el modelado del sistema necesita del modelado del resto de actores ya que este teniendo en cuenta los objetivos, las restricciones y las proposiciones de valor del resto de los actores deberá poseer una serie de tareas y objetivos que se alineen con ellos. El resto de actores exponen el problema mientras que el sistema ofrece la solución.

En primer lugar, se procede a definir las tareas que contribuirán al cumplimiento de los distintos objetivos, utilizando los enlaces de descomposición de tipo OR para establecer alternativas. Estas tareas deben expresarse en el máximo nivel de abstracción posible ya que serán refinadas en fases posteriores del proceso de desarrollo. Una tarea del actor sistema debe estar unida a:

- Un objetivo u objetivo soft de otro actor mediante un enlace de dependencia o contribución.
- Otra tarea mediante un enlace de descomposición.
- Un objetivo u objetivo soft del sistema mediante un enlace de contribución o descomposición.

Las tareas representan los aspectos funcionales del sistema. Para representar los aspectos no funcionales se utilizarán objetivos u objetivos soft (según pertenezca).

De igual manera que en el resto de actores, deberán modelarse las relaciones internas entre los distintos elementos intencionales del actor sistema.

Los elementos intencionales no tienen importancia asignada, esta será asignada según el nivel de importancia de los elementos intencionales a los que estén relacionados mediante enlaces.

### **3.3.4 Discusión y validación**

El modelo de objetivos es validado por un representante de cada actor que verifica que los elementos que aparecen reflejan las características de la organización y del sistema que se desea realizar en su seno, se buscan elementos no incluidos, eliminan redundantes o se modifican aquellos incorrectos. Si existieran cambios estos se incorporarán al modelo hasta obtener el consenso.

Algunas directrices para la validación:

- Un elemento intencional del sistema debe estar conectado con al menos otro elemento intencional de alguno de los otros actores para ser considerado válido.
- No se especificarán tareas de bajo nivel y primarán los objetivos (hard y soft) por encima de las operaciones en el actor principal y los actores externos.
- Las tareas especificadas deben corresponderse a procesos de alto nivel.

## **3.4 Implementación de los métodos de priorización**

Para el presente trabajo el proceso de priorización debe cumplir con las siguientes propiedades:

- **Ser ligero:** El proceso iterativo hace que estas técnicas se empleen repetidas veces en un contexto ágil, una técnica demasiado pesada podría corromper la aproximación ágil. Además, es relevante el esfuerzo de implementación de estas técnicas para el equipo de desarrollo que desee implementar el método.
- **Ser efectivo:** Las técnicas elegidas deben ser capaces de establecer unos parámetros que den lugar a una priorización de los elementos satisfactoria.
- **Integrable a lo largo del proceso de desarrollo:** Dado que el conocimiento sobre el problema aumenta según se desarrolla el proyecto este afectará a las estimaciones que existan sobre el producto por lo que las actividades de priorización deben refinarse según el proyecto avance.

Cómo se ha visto en la descripción del proceso general, la actividad de priorización está dividida en dos actividades: la priorización a nivel de objetivos y la priorización a nivel de tareas. Los motivos de esta división en la priorización en distintas granularidades es debido a que para seguir los principios ágiles y evitar realizar un diseño completo del sistema (propenso a grandes modificaciones según el conocimiento del problema aumente) debe existir una priorización de unidades a alto nivel de manera que se comience el diseño de las unidades más relevantes de manera preferente, sin necesitar detallar todas las actividades, tarea que consumiría muchos recursos tanto en el momento de diseño como re-trabajo futuro en el caso de modificaciones o errores de diseño detectados en fases avanzadas del proyecto.

La priorización a nivel de tareas trata de, una vez detalladas las actividades de alto nivel que cumplirán con los objetivos en tareas de más bajo nivel, establecer prioridades para la selección de las tareas a implementar en la iteración.

A continuación se explican detalladamente ambos procesos y las técnicas utilizadas en ambos.

### **3.4.1 Priorización de tareas**

El objetivo de esta priorización es obtener un criterio para seleccionar qué objetivos, y por extensión que actividades de alto nivel, tienen preferencia y por lo tanto deberían ser desarrolladas en primer lugar. Este proceso toma como entrada el modelo de valor en GRL del cual se procederá a estimar la importancia los objetivos del actor principal y los actores externos que lo forma. Tomando este parámetro como referencia se procederá a propagar la importancia por el modelo de manera que los elementos intencionales del actor sistema queden priorizados según la importancia de los elementos del resto de actores a los que contribuyan.

Para ello los elementos intencionales a estimar la importancia serán los objetivos (objetivos y objetivos soft) de los actores externos y actor principal.

La técnica de estimación de la importancia utilizada será el consejo de expertos. Los roles implicados en esta estimación serán mixtos de la organización cliente y del equipo de desarrollo y en caso de ser posible de los actores externos, aportando expertos del dominio y analistas.

La técnica de valoración elegida será la de Análisis por Puntos de Valor (*Value Points Análisis*, VPA), ya comentada en la sección 2.6, esta puede verse como una técnica mixta del Juicio de Expertos y el *Planning Poker*. Esta técnica combina la sencillez de

aplicación con la participación de los distintos implicados en el proceso de estimación, de manera que se aprovechen los conocimientos de los distintos implicados. La mecánica de la estimación será la siguiente:

- Se valorarán cada uno de los objetivos (objetivos y objetivos soft) de los actores externos y principal.
- En cada uno de estos elementos se llegará a acuerdo entre los representantes de los stakeholders involucrados una puntuación del conjunto de valores discretos numéricos (0, 25, 50, 75,100).
- Una vez valorados todos elementos intencionales, este valor de importancia es utilizado para establecer la lista de objetivos ordenada por importancia.

Emplear la técnica de *Value Points Análisis* en GRL es sencillo ya que el lenguaje soporta este tipo de consideraciones en los elementos intencionales que poseen un atributo Importancia cualitativo o cuantitativo. Este atributo permite valores del 0 al 100 (ambos incluidos) y valores de Alta, Media, Baja o Ninguna respectivamente.

El conjunto de valores discretos que se utilizará es un subconjunto de valores numéricos posibles en el atributo cuantitativo permitido en GRL. Esta elección viene motivada porque un conjunto demasiado amplio de valores complica la valoración y comparación de los ítems (el criterio para diferenciar un ítem con 79 o 81 es difícil de precisar) pero debe ser suficiente para que puedan establecerse rangos suficientes de relevancia para poder clasificar todos los elementos de una manera suficientemente precisa.

La intervención de los stakeholders en la estimación de importancia termina aquí. En muchos trabajos de investigación en la industria se subraya que la intervención de los stakeholders en la priorización es complicada dado que primero, el nivel de abstracción no es el suficiente para que el stakeholder pueda discernir si esa unidad de trabajo es más o menos prioritaria; y segundo porque en muchas ocasiones es complicado tener acceso a los stakeholders. Esta técnica facilita la priorización de elementos que son procedentes del cliente (sus propios objetivos) y por tanto conocidos y de alto nivel y no requiere de la presencia de los stakeholders en posteriores procesos de priorización, aunque los principios ágiles lo consideren recomendable.

A continuación se procede a extender las prioridades de los objetivos sobre las tareas del actor sistema de manera que estas obtengan una valoración correlacionada con la capacidad de estas para cumplir los objetivos de los actores. Esto se realiza de manera autónoma empleando el algoritmo de propagación del valor véase sección 3.4.2.1.

El contexto de la aplicación de técnica es el modelo de GRL. De este modelo no simplemente se extraen los elementos a valorar por su importancia: también están presentes las relaciones de contribución entre sí. Podrían darse casos en los que elementos de importancia media formarían parte de la espina dorsal de varios elementos de importancia superior para la organización, pero que sin embargo a la hora de comenzar a desarrollar de manera iterativa probablemente debieran ceder el paso a ese elemento de, en teoría, importancia inferior.

Añadir los conceptos presentes en los enlaces a la técnica de VPA es una forma poco costosa y sencilla de aportar una información de mayor calidad al modelado y corregir defectos de asignación de valores comunes en esta práctica debido a que los diferentes

representantes que intervienen en la asignación cuentan con perspectivas diferentes y los pesos representados se forman a partir del consenso y con un nivel alto de subjetividad.

Como resultado de esta técnica todas las tareas habrán recibido un valor comprendido entre el [-100, 100] que permitirá su ordenamiento siendo las valoradas con 100 las que más contribuyen a los objetivos de valor de los actores y -100 las que afectan negativamente a estos intereses.

Con respecto al resto de técnicas como AHP, vista en la sección 2.6, basadas en la comparación por pares resultan costosas y la precisión que ganan no resulta tan beneficiosa cuando se trata de un nivel tan alto de abstracción. A su vez, la sencillez de la técnica y su similitud con las empleadas por la mayoría de practicantes ágiles hace que sea fácil de incluir en el proceso y de implementar tanto para el personal técnico como el no técnico.

Esta primera priorización permite hacer una criba preliminar al refinamiento de las actividades del negocio, eliminando aquellas que no contribuyan a los objetivos o dejando para otras iteraciones. Sería deseable en este estado poder realizar una estimación del coste, sin embargo, el nivel de abstracción de las tareas todavía es demasiado elevado como para realizar una estimación lo suficientemente precisa para ser relevante.

### **3.4.2 Priorización a nivel de actividad de proceso de negocio**

Una vez han sido seleccionadas, transformadas en actividades de proceso de negocio y refinadas en más detalle. Es necesario llevar a cabo la priorización de las actividades del proceso de negocio de manera.

En la priorización de actividades a este nivel se va a optar por una aproximación paramétrica. Por una parte se tomará la estimación del valor y por otra el coste de las actividades. Este es un método común tanto en metodologías ágiles como tradicionales. La contraposición de coste y valor permite tomar decisiones acerca de qué requisitos deben implementarse y permite estimar la duración de los *sprints* y del proyecto. Para la estimación de los diferentes parámetros y su asignación a los elementos intencionales del actor sistema se utilizan dos técnicas diferentes.

La estimación del valor se corresponderá con la estimación realizada de la importancia de los objetivos del actor principal y los actores externos. Dado que el diagrama GRL une estos objetivos con las tareas mediante relaciones de contribución y/o dependencia es posible extrapolar qué valor aportan las tareas a esos objetivos en base a la contribución que hagan. Dado que una tarea puede contribuir a más de un objetivo (tanto positiva como negativamente) al mismo tiempo, se ha desarrollado un sencillo algoritmo que calcule el valor de cada elemento en base a sus contribuciones (ver en sección 3.3.3.4).

El otro parámetro a tener en cuenta es el coste de las distintas tareas. Dado que el nivel de abstracción de las tareas en este momento de estimación inicial es demasiado elevado para cómo realizar una estimación precisa, se ha optado por realizar una estimación del coste relativa. De esta manera se consigue la priorización entre elementos teniendo en cuenta su coste independientemente del conocimiento de su coste real. Como se ha visto en la sección anterior, la precisión de la estimación varía en el tiempo de desarrollo, por

lo que en este trabajo se propone una estimación iterativa, que sea precisada según las necesidades de la fase de maduración del proyecto.

Teniendo en cuenta las técnicas vistas en la sección 3.3.2, para la estimación de las tareas especificadas en el diagrama de valor en GRL, se utilizará la técnica de juicio de expertos bajo la forma de *planning poker*. Esta técnica es útil a este nivel de abstracción del proyecto debido a que la estimación no necesita de gran precisión y a la vez es una técnica ágil y sencilla de implementar. Esta técnica permite tener en cuenta los conocimientos previos de los implicados en el desarrollo así como de los datos previos de otros desarrollos similares para realizar la estimación.

Para ponderar el peso del coste frente al valor que se presta existen muchas técnicas, basándose en el retorno de inversión, en este caso se aplicará:

$$\text{Prioridad} = \text{Valor estimado} / \text{Coste estimado}$$

Se ha elegido esta técnica por razones de simplicidad pero pueden utilizarse otras ponderaciones (considerando el valor más relevante que el coste o en otro porcentaje) o incluso realizarse la priorización conforme a qué objetivos contribuye cada una de las actividades. Esta elección dependerá de la organización pudiendo seleccionar la práctica que les de mejores resultados o la que consideren oportuna en cada desarrollo.

#### **3.4.2.1 Algoritmo de propagación de la importancia**

Para la propagación de la importancia en un algoritmo de GRL se emplea este sencillo algoritmo que pondera las contribuciones de cada elemento para calcular la importancia relativa del mismo. Este algoritmo trata de valorar el impacto de las contribuciones de los elementos intencionales, de manera que el valor de un elemento intencional aumenta (de manera proporcional a la importancia del elemento al que contribuye y la medida en lo que lo hace) si este contribuye positivamente a otros elementos intencionales y, de la misma manera, disminuye si contribuye de forma negativa a otros elementos intencionales. De esta manera se obtiene un valor relativo de cada elemento intencional.

El algoritmo se aplica en dos fases, en primer lugar, sobre los actores externos y el actor principal, cuyos elementos están ya valorados de manera que este algoritmo añade la información relativa a sus contribuciones a su valor cuantificado. Para ello se procede de la siguiente manera:

Para cada elemento intencional de las tareas del actor sistema:

1. Tomar todos los elementos intencionales del actor
2. Para cada uno calcular:

$$\text{Valor}(E) = E_{\text{importancia}} + \sum_{i=0}^{Ec} \left( \frac{Ec_i}{100} * \text{Valor}(Eco_i) \right) + \frac{1}{E_h} * E_{p_{\text{importancia}}}$$

Dónde:

E = Elemento Intencional



$E_{importancia}$  = Importancia del elemento intencional

$E_c$  = Conjunto de todas las contribuciones del elemento intencional.

$E_{c_n}$  = Contribución n.

$E_{o_n}$  = Elemento intencional objetivo de la contribución n.

$E_p$  = Elemento intencional padre.

$E_h$  = Número de elementos intencionales hermanos.

$E_{p_{importancia}}$  = Importancia del elemento intencional padre.

Una vez calculados estos se procede a aplicar el algoritmo sobre los elementos intencionales del actor sistema de manera similar. Con la diferencia de que estos elementos no se hallan valorados directamente, sino que su valor depende de las contribuciones que hagan a los elementos intencionales del resto de actores, es decir, en la medida que son útiles al resto. Se procede de forma similar utilizando la función Valor anteriormente descrita:

1. Para cada elemento intencional del actor sistema calcular:

$$Valor_{sistema}(E) = \sum_{i=0}^{E_c} \left( \frac{E_{c_i}}{100} * Valor(E_{c_{o_i}}) \right) + \frac{1}{E_h} * E_{p_{importancia}}$$

Este algoritmo utiliza los valores de contribución de los enlaces de manera similar a como las estrategias de GRL los utilizan para realizar simulaciones de valor [23]. En este caso los enlaces de dependencia, representan contribuciones de valor 100 (convención habitual para la implementación de estrategias en la herramienta de GRL jUCMNav).

En el caso de las descomposiciones, se ha optado por la simplificación de descomponer en partes iguales el valor del elemento padre, de esta manera, todos sus elementos hijos tienen la misma prioridad, ya que probablemente se elijan para formar parte del sprint juntos pero posean el valor representativo suficiente para poder ser priorizados entre ellos y entre el resto de elementos.

### **3.5 Uniendo procesos de negocio con la especificación del valor**

#### **3.5.1 Selección de un lenguaje de representación de los procesos de negocio**

El modelo de negocio en el proceso de desarrollo orientado al valor del cliente tiene el papel de representar con un nivel de abstracción alto los requisitos funcionales definidos por el modelo de valor en GRL ordenados en una secuencia de actividades de negocio.

El uso de modelado de proceso de negocio viene motivado por la necesidad de primero comunicar al cliente el efecto del sistema de información sobre su organización, la funcionalidad que este le va a ofrecer y su integración con los procesos existentes. En este caso el modelo es herramienta de comunicación para que el cliente pueda validar desde

el inicio la especificación de su sistema sin que este tenga que estar desarrollado o modelado por completo, en una representación entendible por personal no técnico.

Por otro lado este modelo permite la planificación de la arquitectura y del desarrollo de las iteraciones dado que en primer lugar, refina la información elicitada en el modelo de valor a la cual añade valores de secuencia y concretar objetivos como actividades de negocio.

De esta manera y usando su relación con los elementos de valor es posible dividir el sistema, eligiendo aquellas unidades que aporten mayor valor al cliente como principales candidatas a ser integradas en una iteración de desarrollo y por lo tanto ser entregadas primero.

El uso de objetivos de proceso tiene como finalidad alcanzar un nivel más alto de abstracción en los modelos de proceso empleando los objetivos para describir qué se debe conseguir en vez de como se debe realizar. Los medios para conseguir este fin entonces pueden ser expresados en un nivel de detalle mayor utilizando un flujo de trabajo tradicional orientado a las actividades.

Ventajas:

- **Ampliamente aceptado y conocido:** Hace más sencillas las conversaciones con los clientes y la validación de los modelos ya que es probable que estén familiarizados con la notación. De la misma manera, si existen en la organización procesos de negocios modelados, con casi total seguridad estarán expresados en esta notación. También han sido investigados con profundidad lo que permite más versatilidad a la hora de realizar modelos.
- **Granularidad/nivel de abstracción:** Los tipos de modelos a realizar con BPMN permiten cualquier nivel de abstracción, desde el más alto, representando sólo las actividades más relevantes sin expandir, hasta flujos de procesos extremadamente detallados hasta el nivel más operacional.

Desventajas:

- **Integración con GRL:** El tipo de enlace entre los modelos de GRL y un modelo de BPMN tendrá que ser indirecto o bien recurriendo a un modelo intermedio que realice los enlaces entre ambos o bien especificando como propiedad el elemento origen de cada elemento del modelo de proceso de negocio con su componente en GRL.

Ventajas:

- **Integración con GRL:** Al formar parte del mismo estándar elementos de ambos tipos de diagramas pueden estar conectados de manera directa mediante un elemento denominado URN link, facilitando la trazabilidad entre elementos.

Desventajas:

- **Granularidad:** Demasiado operacional. Se tiene a usar como un conector directo entre objetivos y casos de uso, no posee la notación necesaria para tratar

de englobar elementos a más alto nivel que permitan trabajar con ellos a ese nivel de abstracción.

- **Popularidad:** *Use Case Maps* no es una técnica demasiado popular con lo que podría suponer un sobre coste de esfuerzo para incorporar, sobre todo al personal no técnico, en la discusión de los modelos. A su vez complicaría la integración con otras tecnologías en el proceso de desarrollo. Potente pero deja parte de la implementación interesante para este caso en manos del usuario.

En la presente sección se expondrá la propuesta para la derivación de procesos de negocio a partir del modelado de valor. En primer lugar, se explorará el estado del arte en esta cuestión y en segundo se propondrá una técnica que en el contexto de la metodología propuesta permita guiar la conversión de los objetivos del modelo de GRL en procesos de negocio.

### **3.5.2 La unión de modelos de valor y de procesos de negocio en la literatura**

En la literatura se ha investigado la relación conceptual entre objetivos y diseño de proceso de negocio y se han propuesto algunas guías para el paso de los objetivos al proceso de negocio. No solamente se trata de una conversión entre modelos, sino que es un salto conceptual que no puede ser completamente automatizado ya que consiste en derivar objetivos (el valor del cliente) en los procesos que permiten explotar ese valor.

En este trabajo se requiere analizar el modelo de valor de GRL para identificar los procesos de negocio necesarios para aportar valor al cliente según lo expresado en el modelo. Por ello se buscan en la literatura guías o procesos que permitan facilitar esta transformación y a su vez permitan establecer una conexión entre los propios elementos sintácticos que definan los procesos de negocio y el modelo de valor expresado en GRL.

En la tesis [71] aunque orientado hacia la monitorización de procesos basada en los objetivos, sus reflexiones sobre las relaciones entre los dos modelos son igualmente relevantes para el presente problema.

En este caso se emplean los mecanismos de URN para unir los elementos de un modelo de Objetivos de Negocio, definido en GRL, con el modelo de procesos de negocio especificado en *Use Case Maps* (UCM, ver sección 2.2), para ello aprovecha que ambos diagramas forman parte del mismo estándar para emplear un tipo de enlace creado para este propósito llamado URN Link que permiten enlazar dos elementos cualesquiera pertenecientes al estándar.

En cuanto a la identificación de los objetivos y cómo estos se trasladan al modelo de proceso de negocio, el autor no especifica ninguna guía o método para realizarlo debido a que el nexo principal de unión es el uso de KPIs que miden un objetivo y un proceso de negocio asociado, dando esta relación como conocida por el modelador debido a su experiencia en el dominio.

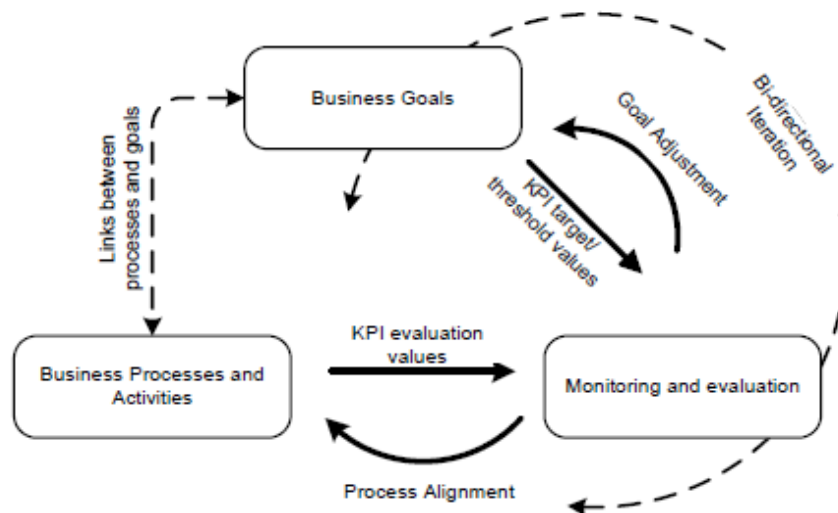


Figura 29 Vista general del proceso de la tesis doctoral de [71].

En [71] se emplean una ontología creada por ellos mismos en la que definen los objetivos y los elementos necesarios para realizar el razonamiento sobre ellos. Su estrategia para identificar los objetivos que deben contribuir como tareas es descomponer los objetivos de alto nivel en otros de una granularidad menor hasta que corresponden directamente con actividades de negocio de alto nivel que luego pueden ser descompuestas en subprocesos. Aunque conceptualmente correcta se echa en falta algunas guías que ayude al diseñador en su tarea o que posibiliten la semi-automatización del proceso. Solamente alguna regla para la validación de la conversión como que un modelo correcto no debería tener ningún proceso de negocio que no contribuya a ningún objetivo, pero sin llegar a presentar las reglas formalmente o tener un conjunto adecuado para su puesta en práctica.

Otra aproximación de modelado de Proceso Orientado a los objetivos (Go4Flex) [73] utiliza también una versión propia del modelado de objetivos basada en el concepto de *Declarative Goals* (Objetivo declarativo). Un *Declarative Goal* expresa la decisión por la que se ejecutaran actividades del proceso de manera que ayuda a comprender el contexto del proceso. En Se establecen dos modelos diferentes el *Goal Model* (adaptado de BDI, *belief-desire-intention agents*) y el plan Model (BPMN) uno para modelar los objetivos y el otro para modelar el proceso. El diseñador comienza modelando el proceso que modela los nodos hoja (de más bajo nivel) del modelo *Goal Model*. Estos objetivos se clasifican en cuatro categorías: *achieve*, *maintain*, *query* y *perform goals*. Usando una técnica similar al caso anterior, modela hasta obtener unos objetivos altamente detallados a los cuales se les asocian los procesos que les permiten cumplir dichos goals.

Uno de los inconvenientes de esta propuesta es el empleo de un lenguaje de modelado personalizada y fuera del estándar (existiendo GRL) no contribuye a que sea fácilmente implementable e interoperable.

A su vez, aparte de la descripción semántica de los elementos, tampoco existen guías o asociaciones que ayuden al diseñador a relacionar los elementos del modelo de objetivos y del de negocio. Por último, la forma de mantener la trazabilidad entre modelos tampoco es evidente a juzgar por los trabajos reportados.

Relacionado con  $i^*$ , en [73] recogen 9 aproximaciones distintas a la transformación de modelos de  $i^*$  en modelos de proceso de negocio, analizadas en busca de desafíos relacionados con el área. De este análisis se concluye:

- **Falta asociar un rol responsable.** En las metodologías presentadas no se menciona qué rol debe hacerse cargo de las transformaciones.
- **Asociaciones conceptuales insuficientes.** No existen guías suficientes que especifiquen la relación entre los elementos del modelo de  $i^*$  con los elementos de proceso de negocio.
- **Informalidad del algoritmo de transformación.** Los métodos estudiados no utilizan algoritmos de transformación formales.
- **No dar soporte completo a la estructura organizacional.**
- **Falta de comprobaciones de consistencia inter-modelo.** No existen reglas de comprobación para validar la consistencia entre los modelos de  $i^*$  y de proceso de negocio.

El método de [75] se propuso para desarrollar flujos de trabajo seguros partiendo de los análisis tempranos de requisitos para servicios SOA. En este trabajo se utilizan en primer lugar modelos de objetivos utilizando las especificaciones de seguridad del dominio en un lenguaje de modelado propio denominado *Secure Tropos*, estos modelos son utilizados para capturar los actores, las dependencias funcionales, autorización y los problemas de confianza. Estos modelos se transforman en modelos *Secure BPEL* utilizando reglas de asociación de alto nivel. Lamentablemente estas reglas son específicas para el dominio de la seguridad y el modelo de *Secure Tropos*.

En [76] se realiza la transformación en el contexto de la verificación de servicios web dirigida por requisitos. En este método se emplea el lenguaje de modelado  $i^*$ . El proceso que emplean es el siguiente: En primer lugar se modelan los objetivos de alto nivel de un actor y se refinan en sub-objetivos que eventualmente se operacionalizan en tareas de alto nivel. En segundo lugar, estas tareas de alto nivel se descomponen en sub-tareas más detalladas añadiendo la notación del flujo de control al modelo de objetivos. En tercer lugar, se añaden los mensajes que describen las interacciones entre actores y se asocian a las tareas básicas. Por último el modelo de objetivos gráfico se representa formalmente por el lenguaje Formal Tropos que es transformado en BPMN utilizando reglas de asociación de alto nivel.

En [77] los autores proponen el uso de  $i^*$  para modelar modelos de proceso de negocio a partir de modelos de objetivos y a su vez descubrir objetivos a partir de modelos de proceso de negocio. Para transformar los modelos de objetivos en  $I^*$  al modelo de proceso de negocio en BPMN primero los actores son identificados en un modelo de objetivos. A continuación los actores de la organización y externos se asocian a las piscinas de BPMN, donde los actores internos se convierten en calles de las piscinas. Después, un analista de negocio debe decidir la conversión de las tareas del modelo de objetivos que pueden ser convertidas en un subproceso o en una actividad atómica. A continuación, se añaden las notaciones de control y datos al modelo de BPMN y finalmente se modelan los detalles de los subprocesos que se dejaron abiertos. A su vez propone mecanismos para conservar la correlación entre los modelos (en ambas direcciones) en caso de que alguno de ellos fuera cambiado. Sin embargo, a pesar de que el conjunto de reglas sistemáticas son claras, son poco efectivas y gran parte de las decisiones de transformación se dejan al encargado de realizarla.

Lo y Yu [78] proponen un acercamiento mediante un catálogo de referencia por el cual se pueda derivar diseños orientados a servicios a partir de modelos de negocio de alto nivel. Primero se elige un modelo de negocio de referencia a partir de un catálogo y las limitaciones se evalúan en relación con el modelo deseado, modificando si es necesario ese modelo de referencia. En segundo lugar, se instancian los modelos de  $i^*$  a partir de los modelos de referencia y son refinados para cumplir los requisitos específicos del caso en cuestión. En tercer lugar, se extraen patrones de servicios de negocios de este modelo de  $i^*$  que ilustran un modelo de servicio de negocio habitual. En cuarto lugar, los patrones se refinan en procesos de negocio de alto nivel (diagramas de colaboración). No se menciona en el trabajo cómo extraer e importar al modelo de negocio la información del modelo de  $i^*$  ni como se establece la trazabilidad entre los elementos de los modelos. Tampoco se proponen guías para llevar a cabo este proceso que queda a juicio del diseñador. Sin embargo es interesante el uso de modelos de referencia para fomentar la reutilización de modelos, patrones y diseños ya conocidos y válidos.

### **3.5.3 Una guía para la transformación del modelo de objetivos en modelo de proceso de negocio**

Cómo se ha observado en la sección anterior el estado del arte en la transformación de modelos de objetivos a modelos de proceso de negocio no ha alcanzado todavía el punto de madurez, principalmente debido a que es difícil crear guías de transformación completamente objetivas dado que el proceso de transformación en algunos niveles requiere de la experiencia o del aporte personal de un analista.

En esta sección se recogen algunas guías para facilitar el proceso de transformación para el método planteado en el capítulo 3.

Como punto de partida de la transformación se identifican los modelos inicial y destino. En este caso el modelo inicial será el modelado de valor en GRL, con especial foco en el actor sistema ya que es el que contiene la operatividad del sistema de información. El modelo objetivo es un modelo de proceso de negocio de alto nivel de abstracción expresado en BPMN que se irá refinando a lo largo del proceso.

En primer lugar se identificarán los distintos actores que formarán parte del contexto del sistema y que por tanto se transformarán en piscinas o calles de estas en el modelo de BPMN. Un actor estará incluido en el sistema y por tanto en el modelo de proceso si este actor tiene una relación directa con el sistema, es decir si existe una unión mediante un enlace (de dependencia o de contribución) con un elemento intencional del actor sistema. Así mismo, se identificarán (si los hay) aquellos actores que sean partes internas de otros, que serán representados como calles dentro de la piscina del actor principal o si es necesario hacer esta división dentro de uno de los actores modelados.

A continuación se deben identificar qué actividades de proceso de negocio se deben extraer del modelo de GRL. Para ello los elementos intencionales modelados como tareas son un claro candidato a conformar una actividad de negocio. Cada elemento que pertenezca a un actor será transformado en un elemento incluido en la piscina que representa a ese actor. Los elementos intencionales objetivos u objetivos soft pueden ser operacionalizados en una actividad o en un conjunto de actividades. En algunos casos, sobre todo algunos objetivos soft como la usabilidad, no son fácilmente trasladables a un

diagrama de proceso de negocio, a pesar de su importancia para el producto final, queda en manos de los desarrolladores trasladar estas propiedades al producto final.

Estas actividades en muchos casos estarán en un nivel alto de abstracción y será necesario detallarlas después para hacerlas efectivas, el siguiente paso es identificar aquellas actividades candidatas a ser detalladas ya que efectivamente se tratan de subprocesos.

Una vez identificadas las tareas y subprocesos se deberá introducir los conceptos de flujo y paso de mensajes así como elementos de decisión en el modelo (puertas AND y OR). El modelo de GRL puede ayudar en esta tarea ya que relaciones entre distintas tareas (por ejemplo una descomposición AND puede indicar un flujo que atraviese las actividades descompuestas), dependencias o relaciones de contribución entre elementos intencionales de dos actores puede indicar un intercambio de mensajes o un flujo de trabajo. Aunque es difícil precisar una relación determinista entre ambos.

Los enlaces entre los elementos de los distintos modelos pueden hacerse explícitos en la práctica mediante la propiedad *Metadata* de los elementos intencionales de GRL, en ella se puede incluir un elemento EnlaceBPMN que contenga una referencia al elemento del modelo de BPMN al cual se ha transformado. La trazabilidad entre ambos modelos es una propiedad importante que permitirá en el futuro identificar efectos de cambios, justificar las actividades del proceso de negocio y monitorizar las propiedades de valor expresadas en GRL mediante su implementación como actividades de negocio (para más detalles consultar la sección 6.3 Trabajo Futuro).

Finalmente, se debe verificar la completitud del modelo, verificando que cada actividad aparecida en el modelo de proceso de negocio tiene como referente al menos un elemento del modelo de GRL y viceversa. En caso contrario deberá o prescindirse de esa actividad si un análisis cuidadoso lo cree necesario o modelar esta actividad en el modelo faltante de manera que se conserve siempre la consistencia entre ambos.

## **4. Caso de estudio**

En este capítulo se procederá a realizar una prueba de concepto del método propuesto, incluyendo las tareas del proceso vistas en el capítulo 3, desde la 3.2.1 a la 3.2.7, capturando el núcleo del proceso de modelado del valor y su uso para la priorización de las actividades de negocio. Para ello se utilizará un caso de estudio adaptado de la literatura, sobre el que se aplicará paso a paso el método y se comentarán los resultados obtenidos.

### **4.1 Definición: Agencia de Viajes**

Este caso de estudio es una adaptación del presentado por Valeria Castro en [70].

Como caso de estudio se ha escogido una agencia de viajes que presenta un conjunto de servicios a sus clientes, estos servicios son: alojamiento, transporte y paquetes turísticos, que incluyen ambos, transporte y alojamiento.

El funcionamiento actual de la agencia de viajes es el siguiente: primero el cliente provee una descripción de los requisitos del viaje a la agencia de viajes, incluyendo sus restricciones y preferencias personales. Después la agencia de viajes selecciona al agente intermediario que puede proporcionar este tipo de viaje de acuerdo a las necesidades del cliente. Cada agente intermediario puede trabajar con varias Compañías de Transporte, pidiéndoles que provean con una oferta para el servicio requerido. Entonces, la agencia de viajes ofrece al cliente las diferentes opciones, de acuerdo a las preferencias del cliente. Una vez el cliente selecciona uno de los viajes, él da los detalles de una tarjeta de crédito a la agencia de viajes quien procesa el pago a través de la compañía financiera correspondiente. Después de comprobar que el pago es correcto la agencia de viajes pregunta al agente intermediario correspondiente confirmar la reserva y se notifica al cliente. Finalmente, una vez al mes la agencia de viajes paga a cada agente intermediario externo por los servicios que ha ofrecido en el mes pasado.

Esta agencia de viajes quiere construir una aplicación en la nube para dar soporte a este proceso de negocio que incluya a todos los roles que intervienen.



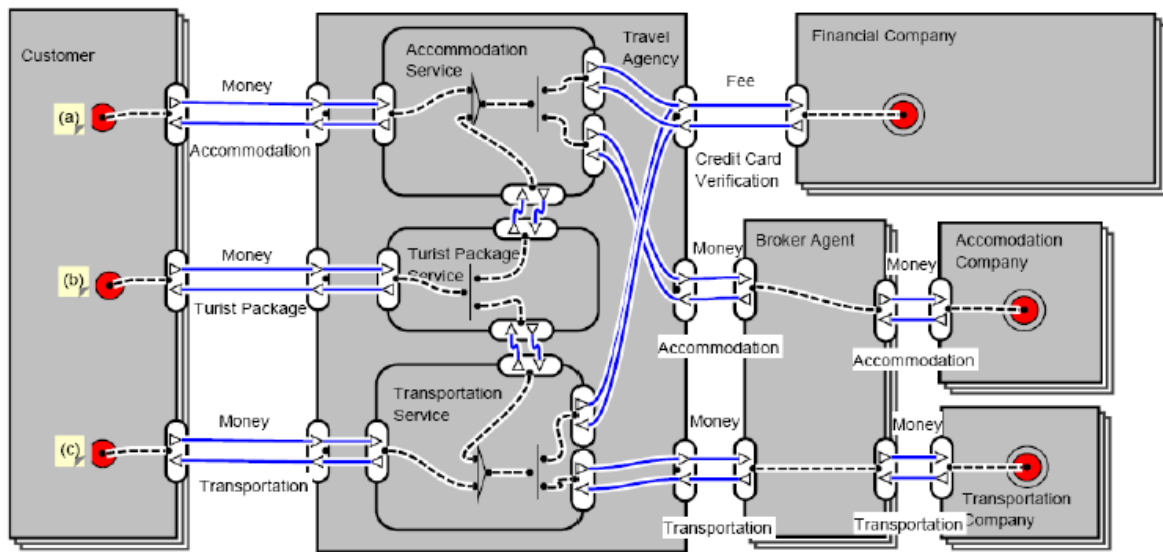


Fig.3 Value Model for the Travel Agency case study

### Figura 30 Modelo de valor e<sup>3</sup>-value

Dado que es complicado obtener información sobre qué aspectos son de valor, sin basarse plenamente en conjeturas, sin acudir a la fuente original, en este caso la agencia de viajes, se ha incluido también como input la representación e<sup>3</sup>-value del flujo monetario de la organización de manera que se tenga más información acerca de las actividades económicas que se realizan y por tanto sea más fácil identificar los elementos de valor y sus relaciones en el modelo de valor.

A continuación se describirán con detalle paso a paso el proceso descrito en la sección 3 de este trabajo.

## 4.2 Aplicación del método de modelado del valor

### 4.2.1 Iniciación

Dado que la intención principal de este caso de estudio es ilustrar el método de especificación del valor propuesto, se prescindirá de los detalles asociados puramente a la gestión de proyectos y se darán como acordados. Por lo que en esta fase solamente se atenderá a la obtención del enunciado del problema.

Atendiendo a la descripción de la actividad de la agencia de viajes de apartado 4.1, el cliente necesita una aplicación en la nube accesible vía web, dado que es el método de más fácil acceso para los clientes, en la que los empleados de la agencia, los clientes, las entidades financieras y agentes intermediarios puedan interactuar para desempeñar sus funciones.

Esta aplicación web debe permitir en primer lugar a la agencia de viajes ofrecer la venta de servicios de alojamiento, paquetes turísticos y servicios de transportes así como a los clientes comprar dichos servicios por medio de pago a través de una entidad financiera. Estos servicios serán provistos por agentes intermedios que introducirán los servicios encontrados en las distintas compañías de alojamiento y transportes. Dado que esta actividad de búsqueda de ofertas en estos servicios no está automatizada (el

intermediario puede contactar de cualquier manera con estas compañías) estas no se incluyen en el sistema, si no que el agente es el encargado de introducir las ofertas encontradas.

## 4.2.2 Especificación del valor

En este paso se procede a realizar el modelado del valor utilizando el lenguaje GRL.

### 4.2.2.1 Definición de actores

Se han identificado seis actores:

- **Agencia de viajes:** Este actor representa a la agencia de viajes cliente y es el actor principal.
- **Entidad financiera:** Entidad financiera que se hace cargo de realizar los pagos entre cliente y agencia de viajes. Es un actor externo.
- **Cliente:** Personas interesadas en la adquisición de un producto de la agencia de viajes. Es un actor externo.
- **Agente intermediario:** El encargado de buscar e introducir los productos turísticos encontrados en la compañía de transportes y alojamientos en la agencia de viajes. Es un actor externo.
- **Compañía de alojamientos:** Proveedor de oferta de alojamiento. Es un actor externo.
- **Compañía de transportes:** Proveedor de ofertas de transporte. Es un actor externo.

Además del actor sistema que representa a la aplicación web en la nube que se quiere desarrollar.

Quedarían representados en el diagrama modelado en la aplicación jUCMNav de la siguiente manera:

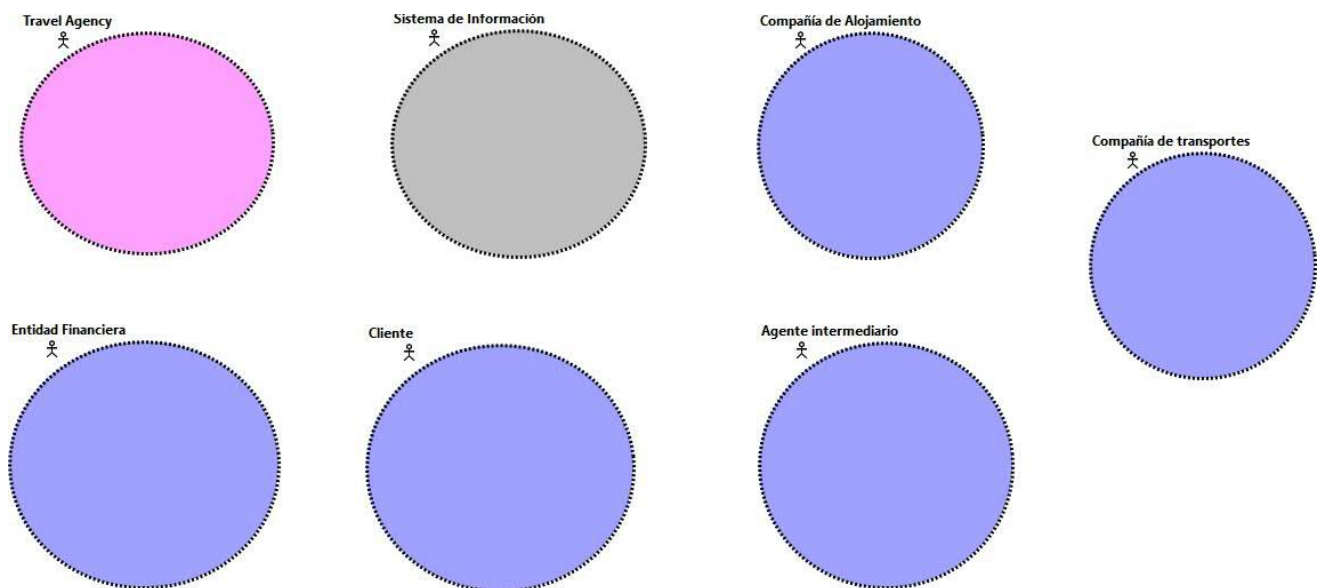


Figura 31 Representación de los actores en GRL

Nota: Los colores escogidos no forman parte del estándar de GRL se utilizan con el propósito de identificar y resaltar con facilidad los elementos del diagrama. Se ha

escogido el color rosa para identificar al actor principal, el azul para identificar a los actores externos y el gris para identificar al actor sistema.

#### 4.2.2.2 Modelado de los actores

##### 4.2.2.2.1 Modelado de los elementos intencionales

Actor Agencia de Viajes:

- **Venta de viajes:** La venta de productos turísticos (transportes y alojamiento) el principal objetivo de la agencia de viajes y la principal fuente de negocio de la organización.
- **Satisfacción del cliente:** La organización persigue que los clientes estén satisfechos con sus compras realizadas a través de ellos ya que los clientes satisfechos suelen volver a comprar.
- **Seguridad:** La organización necesita que las compras se realicen en un entorno seguro de manera que las transacciones monetarias se realicen con éxito y sin incidentes y los datos de los usuarios se almacenen de manera privada.
- **Oferta de viajes:** La organización habitualmente ofrece ofertas de viajes para captar clientes.
- **Fidelización:** La organización está interesada en que los clientes que compran en ella, repitan.
- **Atención personalizada:** La agencia de viajes está interesada en que cada cliente consiga el viaje que más se ajuste a sus necesidades.
- **Satisfacción del Agente Intermediario:** Dado que la calidad de los productos

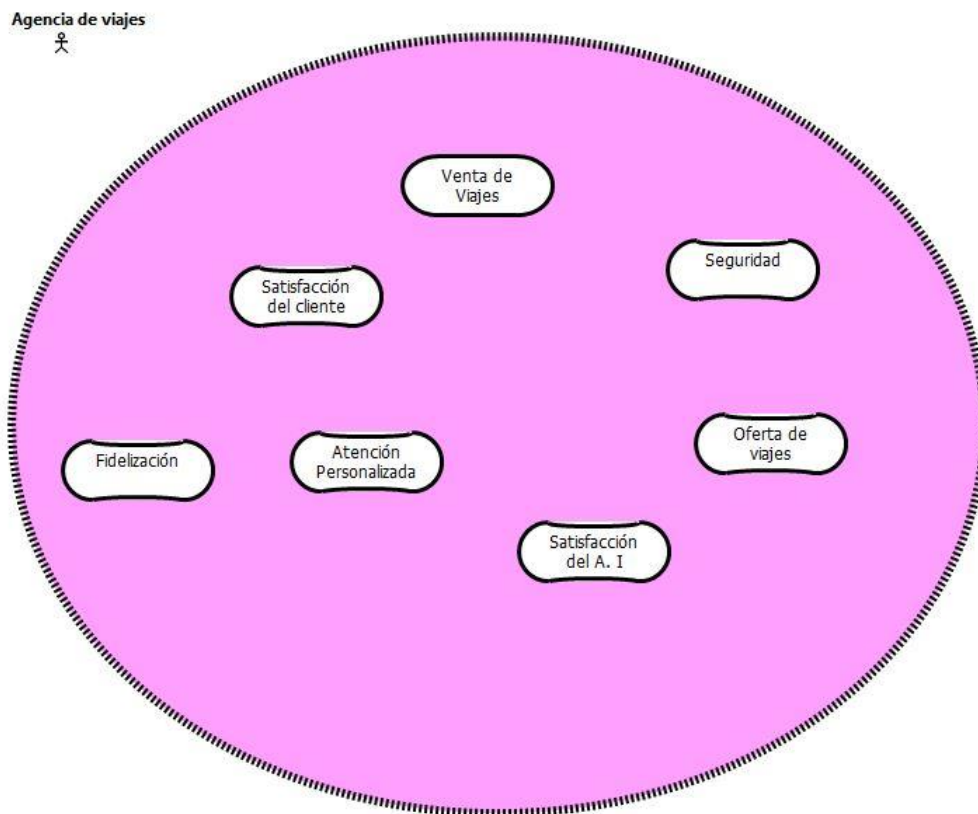
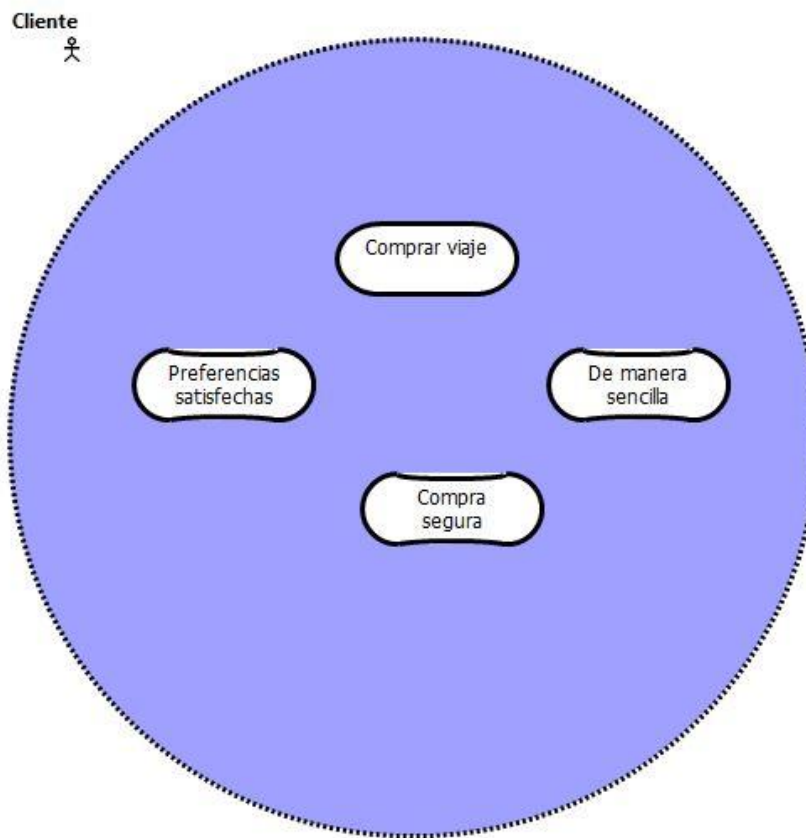


Figura 32 Modelado de los elementos intencionales de la agencia de viajes.

Actor cliente:

- **Comprar Viaje:** El objetivo principal del cliente potencial de la agencia de viajes.
- **Preferencias satisfechas:** El cliente cuenta con un conjunto de requisitos para seleccionar su viaje tales como el precio, localización, calidad, etc. que deben ser cubiertas con el viaje comprado.
- **Sencillez:** El cliente quiere que la compra del viaje a la agencia se haga de la forma menos complicada posible.
- **Compra segura:** El cliente quiere tener la garantía que los pagos que efectúe no tengan ningún riesgo para él.



**Figura 33 Modelado de los elementos intencionales del actor cliente.**

Actor Agente Intermediario:

- **Encontrar el viaje adecuado para el cliente:** El agente intermediario debe seleccionar aquellos productos que se ajusten a las necesidades potenciales de los clientes
- **Obtener ofertas de proveedores:** El principal cometido del agente intermediario es encontrar aquellas ofertas de alojamientos y transportes y añadirlas a las disponibles por la agencia de viajes.

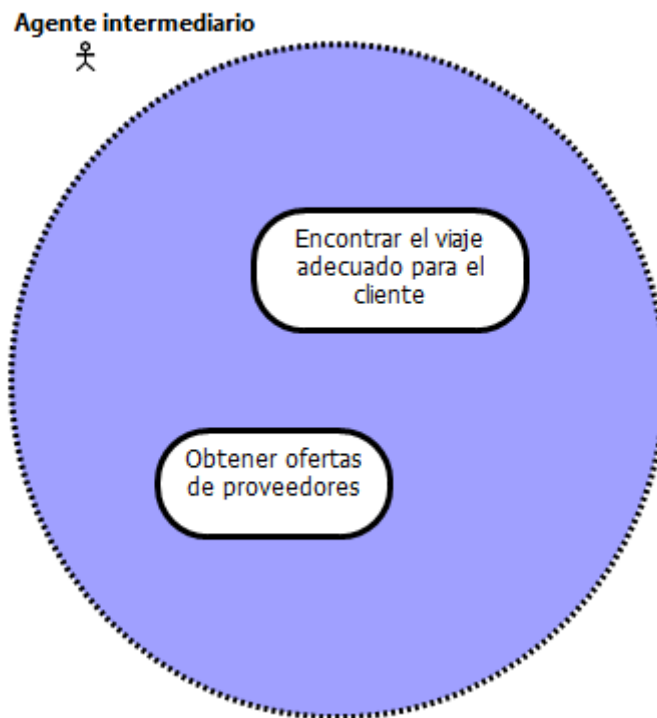


Figura 34 Modelado de los elementos intencionales del actor agente intermediario.

Actor Entidad Financiera:

- **Servicio de pago seguro:** En este caso la entidad financiera solo es relevante para la agencia de viajes y su proceso de negocio para realizar los cobros por las transacciones online.

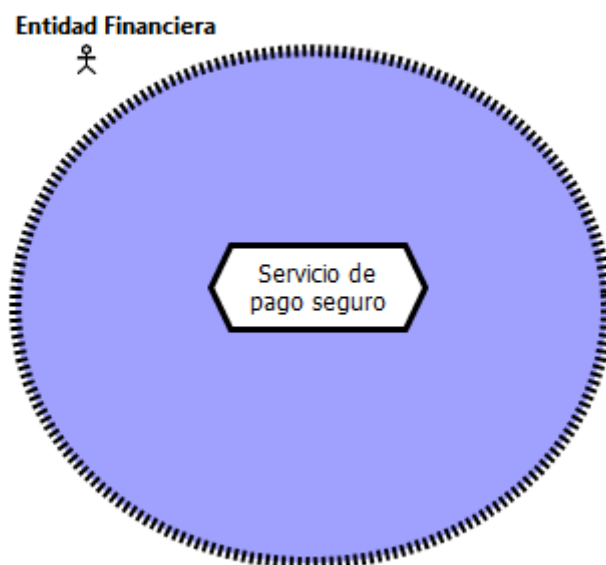


Figura 35 Modelado de los elementos intencionales del actor entidad financiera.

Actor Compañía de Transportes:

- **Ofertar transportes:** Solamente es necesario para este contexto (simplificado) la capacidad de la organización para ofrecer servicios de transporte.

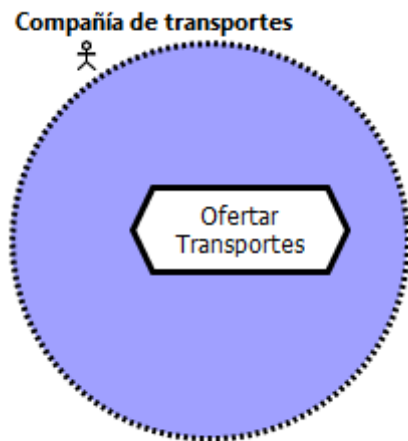


Figura 36 Modelado de los elementos intencionales del actor compañía de transportes.

Actor Compañía de Alojamiento:

- **Ofertar alojamientos:** Solamente es necesario para este contexto (simplificado) la capacidad de la organización para ofrecer productos de alojamiento.



Figura 37 Modelado de los elementos intencionales del actor compañía de alojamiento.

#### 4.2.2.2.1 Modelado de las relaciones y de la importancia

Actor Agencia de Viajes:

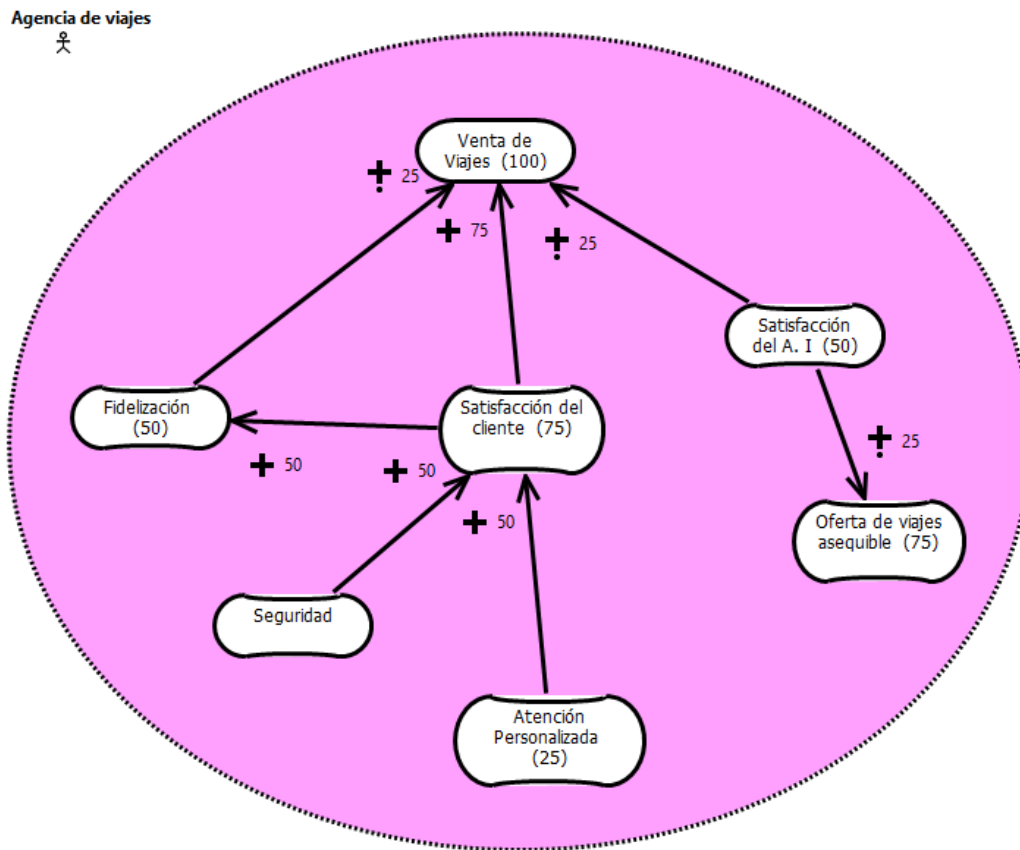


Figura 38 Modelado de las relaciones del actor agencia de viajes.

Actor Cliente

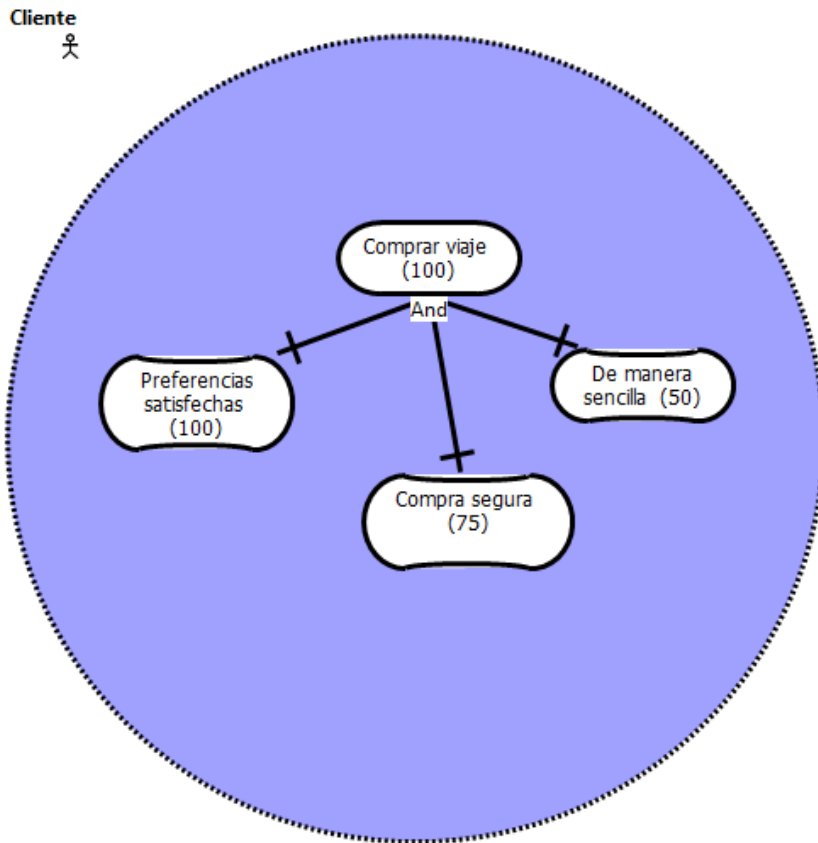


Figura 39 Modelado de las relaciones del actor cliente.



Actores agente intermediario, compañía de alojamiento y compañía de transportes

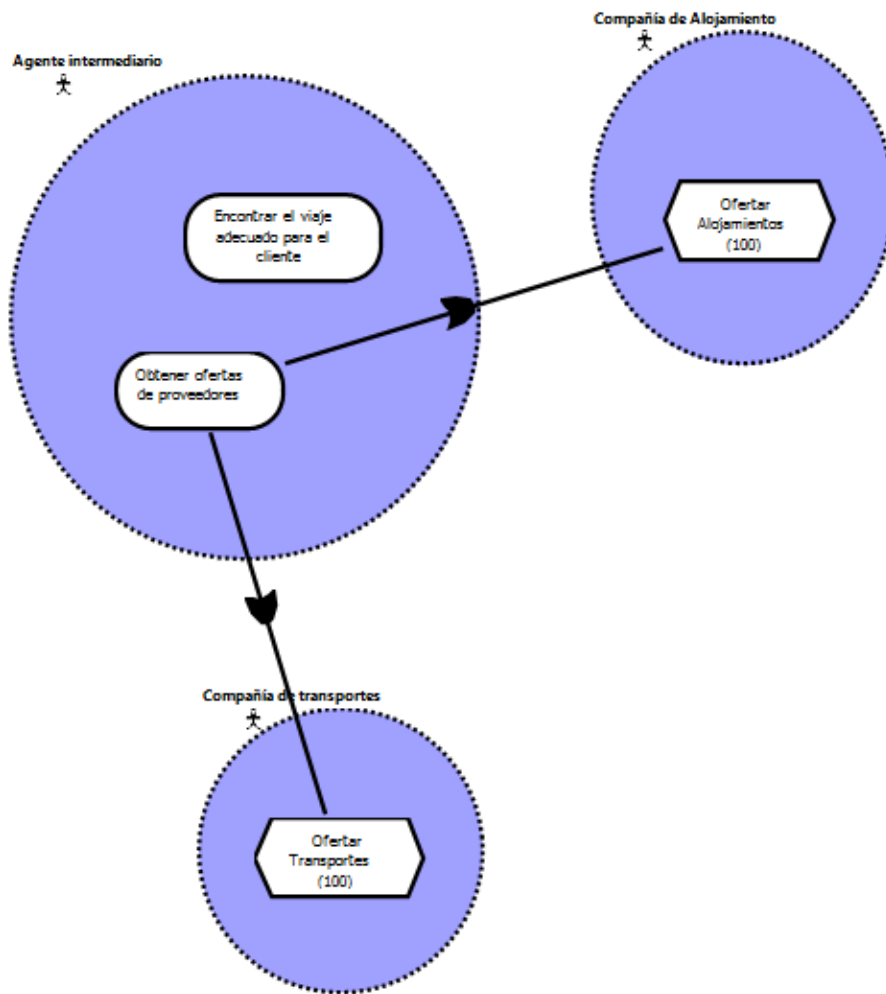
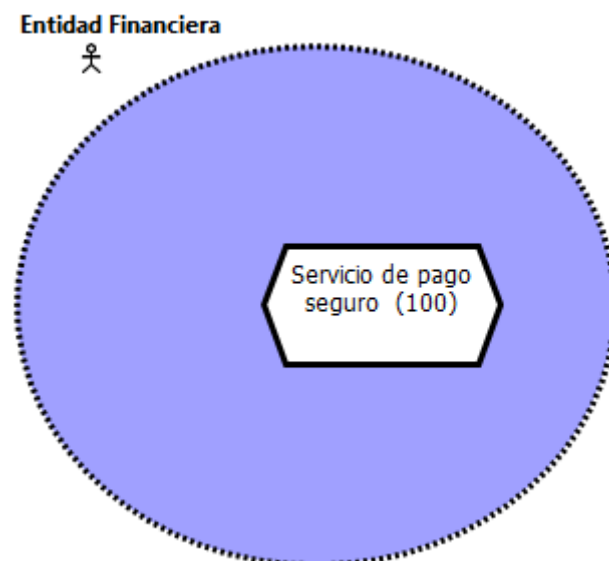


Figura 40 Modelado del agente intermediario y su relación con las compañías de alojamientos y transportes.

Actor Entidad Financiera



**Figura 41 Modelado del actor entidad financiera.**

#### **4.2.2.3 Modelado del actor sistema**

- **Seguridad:** La seguridad del sistema es un aspecto de máxima prioridad para la organización, la aplicación debe ser segura de utilizar, ya que utiliza información sensible, y además esta es una característica crítica para las aplicaciones en la nube.
- **Usabilidad:** Distintos actores valoran que la utilización de la aplicación sea sencilla por lo cual es necesario tener en cuenta esta característica.
- **Disponibilidad:** Con el fin de que los clientes puedan realizar sus compras la aplicación debe de estar disponible lo más cerca posible a la totalidad del tiempo, especialmente en fechas en las que el tráfico se incremente considerablemente.
- **Búsqueda por preferencias:** Para satisfacer los requisitos para la compra de productos de los clientes y ofrecer contenido lo más cercano a estos, se implementará un sistema de búsqueda por preferencias.
- **Chat personalizado:** Con el fin de aportar una atención personalizada a los clientes que lo necesiten se habilitará un chat personal en la aplicación web para que los clientes se comuniquen directamente con la agencia de viajes para resolver cualquier duda o incidencia.
- **Gestión de clientes:** Es necesario que los clientes puedan darse de alta, añadir los datos pertinentes y editarlos; y a su vez que la agencia de viajes pueda utilizar la información de estos para extraer datos de negocio y poder tomar decisiones estratégicas (promociones, mejoras...)
- **Servicio de atención al cliente:** La aplicación permite a los clientes realizar reclamaciones sobre distintas incidencias (técnicas o sobre los productos adquiridos) de manera que los clientes puedan resolverlas de manera sencilla y queden satisfechos con la agencia de viajes.
- **Pago:** El cliente debe poder pagar mediante la aplicación las compras realizadas.
- **Gestión de productos:** Los agentes intermediarios deben de ser capaces de introducir las ofertas de los distintos productos en el sistema, a su vez los clientes deben poder acceder a ellos para adquirirlos. La agencia de viajes también debe poder editarlos.
- **Servicio de venta:** El cliente puede adquirir los productos (alojamiento, transportes o paquetes turísticos) que aparezcan en la web aportando los datos necesarios para ello y realizando el pago a través de la aplicación.

Sistema de Información

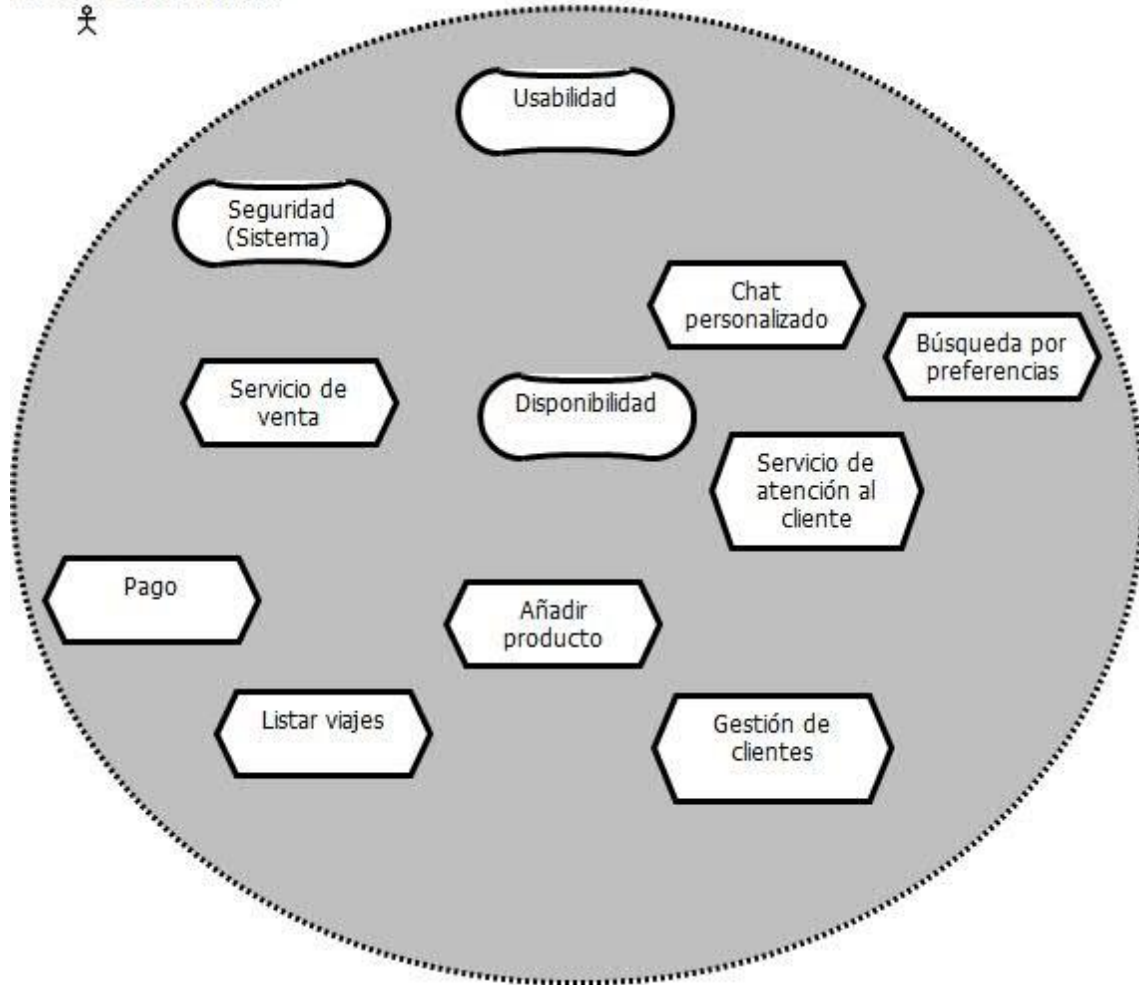


Figura 42 Modelado de los elementos intencionales del actor sistema

A continuación se modelan las relaciones de contribución, dependencia y descomposición propias del actor sistema y con respecto al resto de actores, obteniendo así la representación final en GRL como así se muestra en la Figura 43.

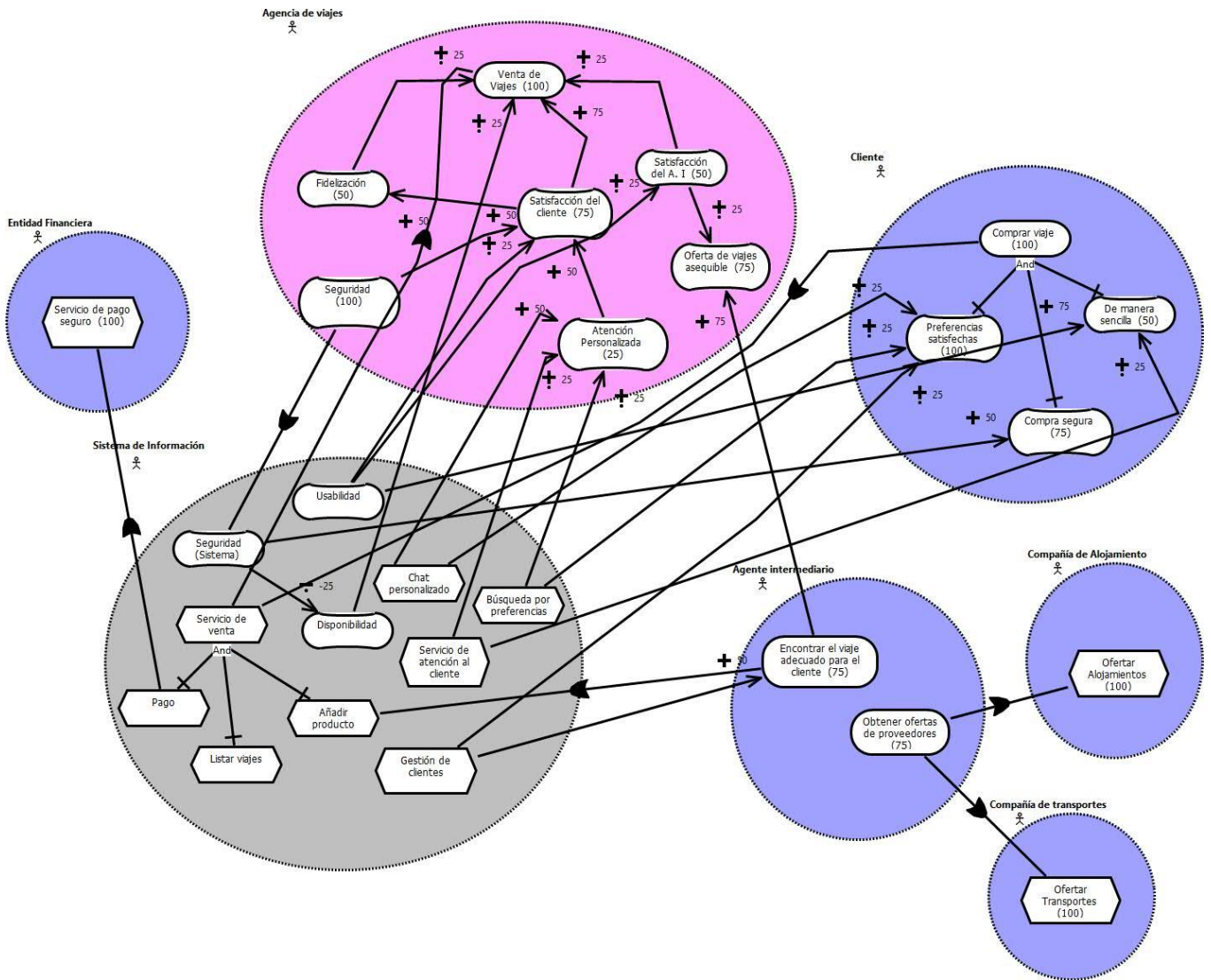


Figura 43 Representación final en GRL.

Debido a las limitaciones de visualización de la herramienta, la complejidad gráfica del ejemplo dificulta su comprensión (a pesar de lo sencillo del mismo). Por ello, para facilitar la comprensión del diagrama, a continuación se detallan en una tabla las diferentes relaciones (de contribución y dependencia, con el valor del enlace designado entre paréntesis) entre los actores con el actor sistema, representados en la figura anterior:

**Tabla 1 Relaciones entre el actor sistema y el actor agencia de viajes**

	Seguridad	Atención personalizada	Satisfacción del cliente	Satisfacción del A.I	Venta de Viajes
Seguridad (Sistema)	Dependencia				
Usabilidad			Contr.(25)	Contr.(25)	
Disponibilidad					Contr.(25)
Chat Personalizado		Contr.(50)			
Servicio de atención al cliente		Contr.(25)			
Búsqueda por preferencias		Contr.(25)			
Servicio de venta					Dependencia

**Tabla 2 Relación entre elementos del actor sistema y el actor cliente**

	Preferencias satisfechas	Compra segura	De manera sencilla	Comprar viaje	Preferencias satisfechas
Seguridad (Sistema)		Contr.(100)			
Usabilidad			Contr.(75)		
Chat Personalizado	Contr.(25)				
Servicio de atención al cliente			Contr.(25)		
Búsqueda por preferencias	Contr.(25)				
Gestión de clientes	Contr.(50)				
Servicio de venta				Dependencia	

**Tabla 3 Relaciones entre agente intermediario y Entidad financiera y sistema**

	Encontrar el viaje adecuado para el cliente	Servicio de pago seguro
Pago		Dependencia
Añadir producto	Dependencia	
Gestión de clientes	Contribución(25)	

### **4.2.3 Priorización de objetivos**

Por las restricciones propias de un caso de estudio, los datos de importancia asignados en lugar de asignarse utilizando los valores de los participantes, se han escogido de manera estimada.

Una vez asignadas las importancias, se realiza una pasada del algoritmo para ponderar la importancia relativa de los objetivos de todos los actores excepto el actor sistema. Esto es, basar su importancia según la asignada y las contribuciones que tengan al resto. Una vez calculadas, el algoritmo de cálculo del valor se utiliza para obtener la importancia relativa de los elementos de valor. Con lo cual se obtienen los siguientes resultados:

**Tabla 4 Importancias calculadas**

<b>Elemento intencional</b>	<b>Importancia calculada</b>	<b>Elemento intencional</b>	<b>Importancia calculada</b>
Seguridad (Sistema)	187.5	Preferencias satisfechas	133.33
Usabilidad	132.81	Compra segura	108.33
Disponibilidad	25	De manera sencilla	83.33
Chat Personalizado	92.71	Comprar viaje	100
Servicio de atención al cliente	50.52		
Búsqueda por preferencias	92.71	Encontrar el viaje adecuado para el cliente	131'25
Servicio de venta	200		
Gestión de clientes	98.96	Servicio de pago seguro	100
Pago	166.67		
Añadir producto	197.92	Seguridad	
Listar viajes	166.67	Atención personalizada	118.75
		Satisfacción del cliente	112.5
Fidelización	75	Satisfacción del A.I	93.75
Oferta de viajes	75	Venta de Viajes	100

**Tabla 5 Elementos intencionales del sistema ordenados por valor**

<b>Elemento intencional</b>	<b>Importancia calculada</b>
Servicio de venta	200,00
Añadir producto	197.92
Seguridad (Sistema)	187.5
Pago	166.67
Listar viajes	166.67
Usabilidad	132.81
Gestión de clientes	98.96
Chat Personalizado	92.71
Búsqueda por preferencias	92.71
Servicio de atención al cliente	50.52
Disponibilidad	25,00

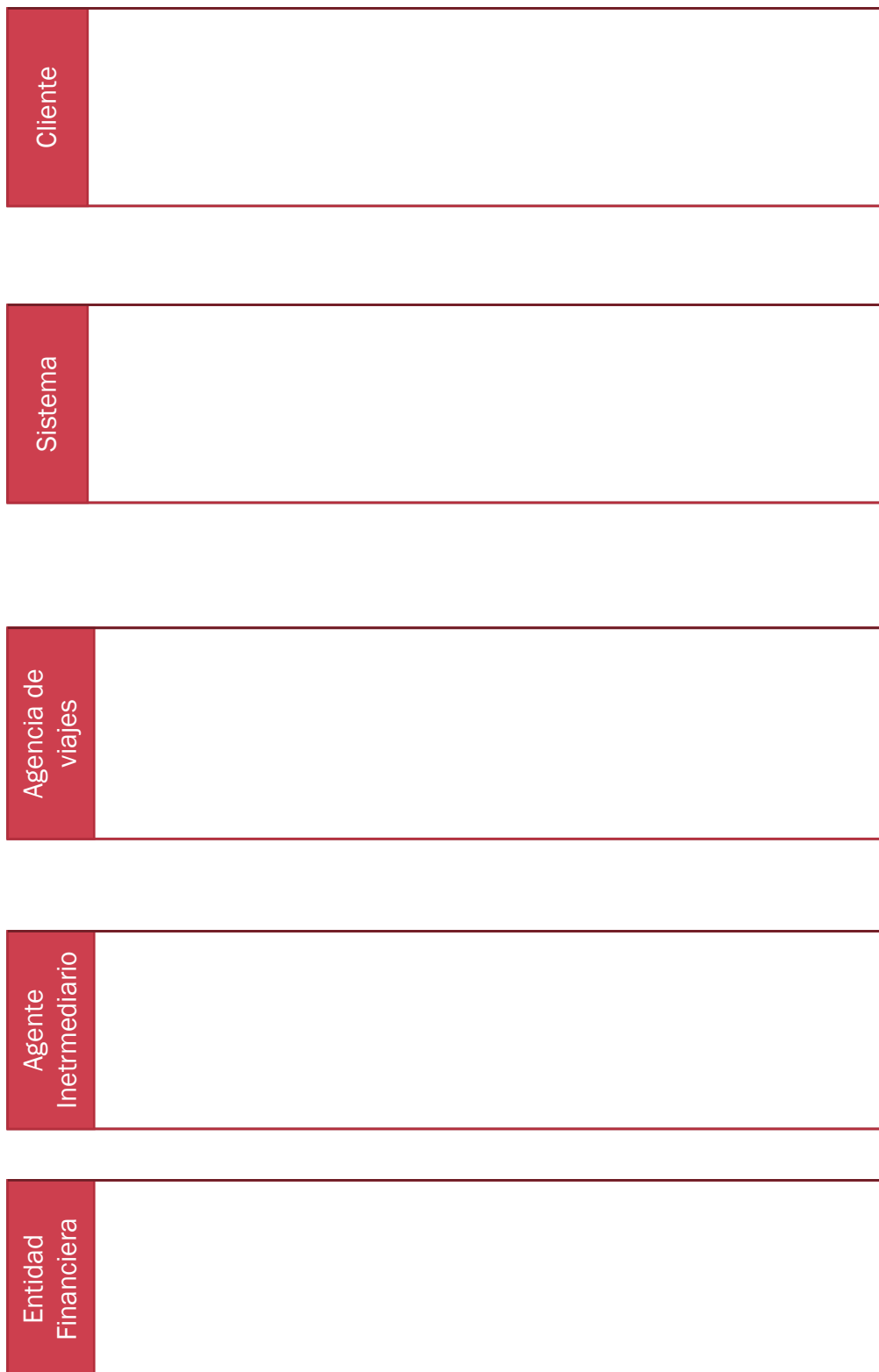
Ordenando los elementos intencionales obtenemos la lista superior. Extrayendo, los elementos intencionales de tipo objetivo soft de los cuáles hemos deducido en principio un orden de prioridad (Seguridad, Usabilidad, Disponibilidad) obtenemos las tareas ordenadas por el valor que aportan al conjunto.

#### **4.2.4 Diseño del proceso de negocio a nivel alto de abstracción**

En esta tarea se toma como partida la descripción del dominio y el modelo de valor en GRL obtenido en el paso anterior para modelar el proceso de negocio en un nivel alto de abstracción en la notación BPMN. Para ello se siguen las guías descritas en la sección 3.5.3

En primer lugar se realiza la transformación de los actores definidos en GRL en piscinas. Se omiten los actores Compañía de alojamiento y @Compañía de transportes dado que estos no tienen dependencias directas con el sistema como se puede comprobar en el diagrama GRL.





**Figura 44 Actores modelados como piscinas en BPMN.**

A continuación se procede a identificar las tareas que se convertirán en actividades del proceso de negocio de cada uno de los actores implicados, añadiéndolas a las piscinas correspondientes y modelando si se trata de un proceso o una actividad atómica.

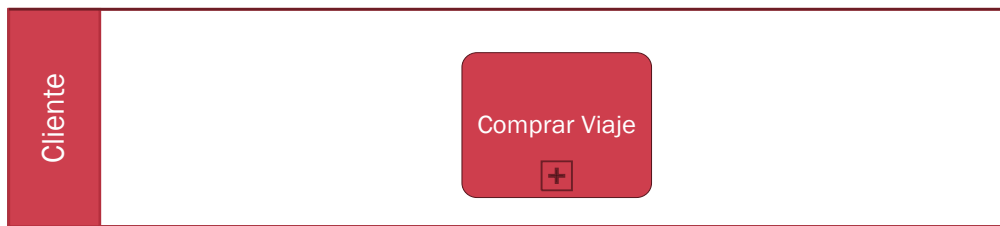


Figura 45 Modelado de las actividades del actor cliente

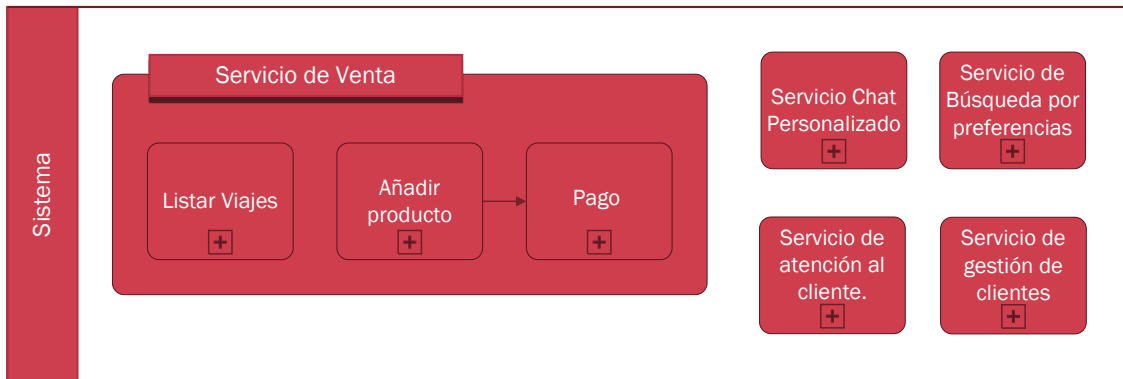


Figura 46 Modelado de las actividades del actor sistema

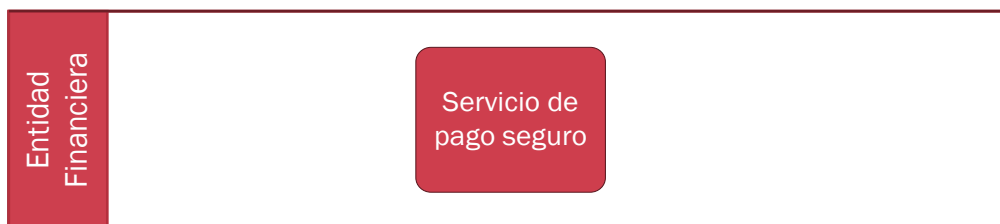


Figura 47 Modelado de las actividades del actor entidad financiera



Figura 48 Modelado de las actividades del actor agencia de viajes

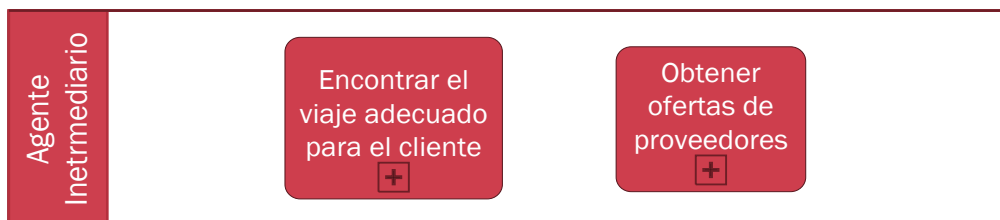


Figura 49 Modelado de las actividades del actor agente intermediario

En la Tabla 6, se detallan las asociaciones entre los elementos de cada uno de los actores de GRL y las actividades que han sido derivadas de ellas. Como se ha comentado en la sección 3.5, la trazabilidad entre ambos elementos se realiza conserva en los metadatos de los elementos intencionales del modelo de GRL, poseyendo un enlace al elemento BPMN al cual ha sido transformado.

**Tabla 6 Asociaciones entre los elementos intencionales de los actores de GRL con las actividades de las calles de GRL.**

<b>Elemento intencional (GRL)</b>	<b>Actividad (BPMN)</b>	<b>Elemento intencional (GRL)</b>	<b>Actividad (BPMN)</b>
Chat Personalizado	Servicio de Chat Personalizado	Comprar viaje	Comprar viaje
Servicio de atención al cliente	Servicio de atención al cliente		
Búsqueda por preferencias	Servicio de búsqueda por preferencias	Encontrar el viaje adecuado para el cliente	Encontrar el viaje adecuado para el cliente
Servicio de venta	Servicio de venta	Obtener ofertas de proveedores	Obtener ofertas de proveedores
Gestión de clientes	Servicio de gestión de clientes		
Pago	Pago	Servicio de pago seguro	Servicio de pago seguro
Añadir producto	Añadir producto		
Listar viajes	Listar viajes	Venta de viajes	Venta de viajes

A continuación se procede a conectar las actividades mediante los flujos de trabajo y de mensajes según sean necesarios.

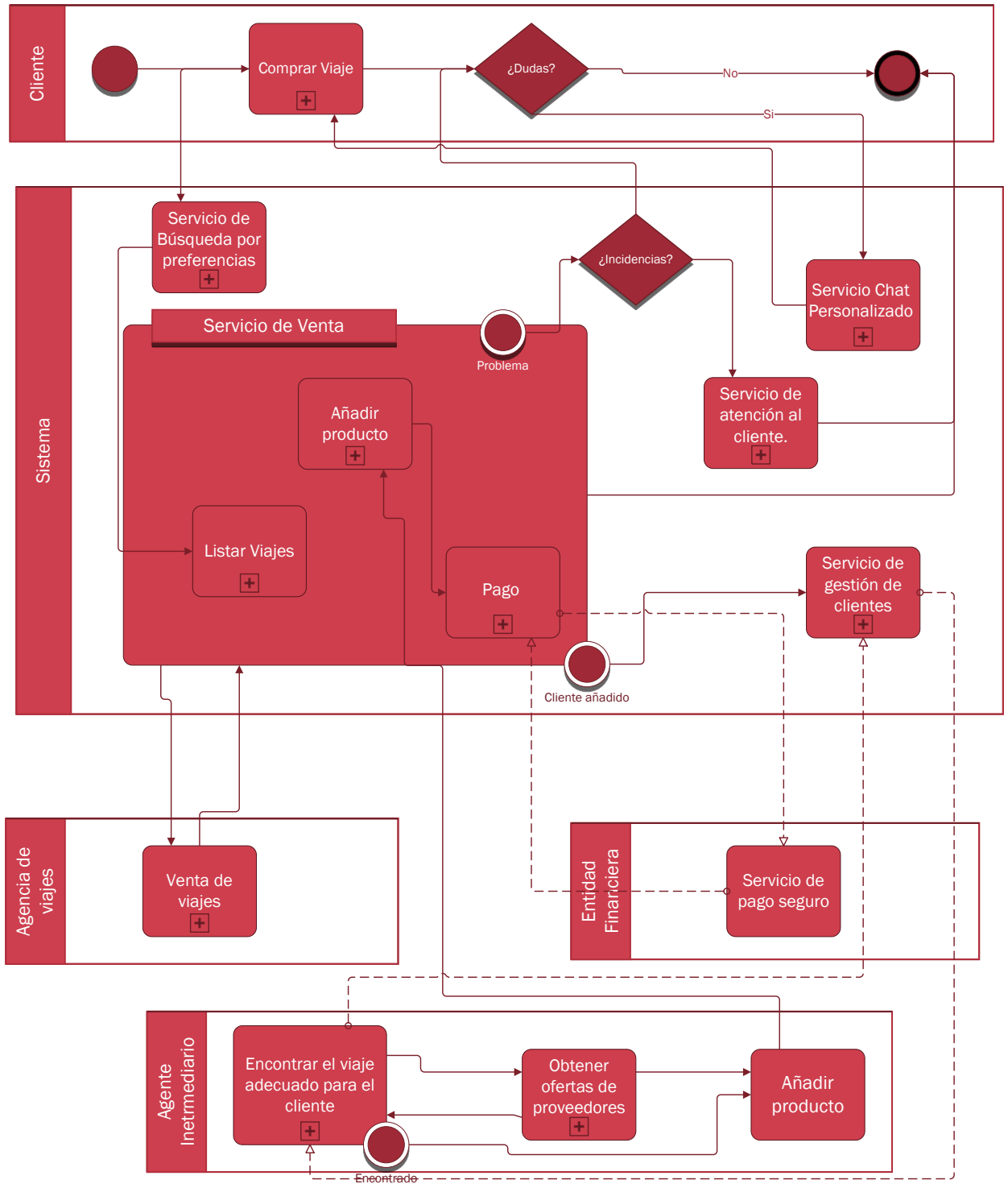


Figura 50 Vista general del diagrama de proceso de negocio en alto nivel de abstracción.

#### 4.2.5 Estimación de esfuerzo

En esta tarea se estima de forma relativa el esfuerzo de implementación de las actividades del sistema. En este caso, los valores obtenidos se han basado en asunciones.

**Tabla 7 Actividades y su esfuerzo relativo estimado.**

Actividad	Esfuerzo
<b>Servicio de búsqueda por preferencias</b>	50
<b>Servicio de ventas</b>	100
<b>Listar Viajes</b>	25
<b>Añadir producto</b>	25
<b>Pago</b>	50
<b>Servicio de gestión de clientes</b>	75
<b>Servicio de atención al cliente</b>	50
<b>Servicio de chat personalizado</b>	25

#### 4.2.6 Priorización de actividades del proceso de negocio

En este paso se lleva a cabo el promedio entre el valor y el esfuerzo de manera que se calcule la prioridad de las actividades. El valor es el obtenido de la tarea del diagrama GRL de la importancia obtenida tras aplicar el algoritmo de propagación de la importancia.

**Tabla 8 Resultado del cálculo de la prioridad sobre las actividades del proceso de negocio.**

Actividad	Valor (Importancia calculada en GRL)	Esfuerzo	Prioridad (Valor/Esfuerzo)
<b>Servicio de búsqueda por preferencias</b>	92.71	50	1,8542
<b>Servicio de ventas</b>	200	100	2
<b>Listar Viajes</b>	166.67	25	6,6668
<b>Añadir producto</b>	197.92	25	7,9168
<b>Pago</b>	166.67	50	3,3334
<b>Servicio de gestión de clientes</b>	98.96	75	1,31946667
<b>Servicio de atención al cliente</b>	50.52	50	1,0104
<b>Servicio de chat personalizado</b>	92.71	25	3,7084

#### 4.2.7 Selección de actividades candidatas para la iteración

Una vez calculada la prioridad, se deben seleccionar aquellas actividades candidatas para ser desarrolladas durante la iteración.

**Tabla 9 Actividades ordenadas por prioridad.**

Actividad	Prioridad (Valor/Esfuerzo)
<b>Añadir producto</b>	7,9168
<b>Listar Viajes</b>	6,6668
<b>Servicio de chat personalizado</b>	3,7084
<b>Pago</b>	3,3334
<b>Servicio de ventas</b>	2
<b>Servicio de búsqueda por preferencias</b>	1,8542
<b>Servicio de gestión de clientes</b>	1,31946667
<b>Servicio de atención al cliente</b>	1,0104

De esta manera se ha obtenido la priorización basada en el valor del cliente y el coste de las actividades de negocio a ser implementadas.

En el supuesto que el proceso de desarrollo del software constara de tres sprints, las actividades a implementar se podrían dividir entre ellos de la siguiente forma:

Al *Sprint 1* le podrían corresponder las actividades añadir productos, listar viajes y el pago, tareas que contribuyen a los objetivos de la venta de viajes, la compra por parte del cliente y la adición de productos por parte del agente intermediario, y los objetivos relativos a la satisfacción del cliente (satisfacción del cliente, fidelización y atención personalizada).

En el *Sprint 2* se completaría la actividad del Servicio de Ventas y el servicio de búsqueda por preferencias (relativo a las preferencias satisfechas del cliente y la atención personalizada).

Por último en el *Sprint 3* se completarían las actividades de servicio de gestión de clientes y servicio de atención al cliente, completando así la contribución a los objetivos del cliente de cubrir sus preferencias y hacerlo de manera sencilla, así como facilitando la tarea del agente intermediario para buscar ofertas de productos deseables por los clientes y por último repercutiendo de forma en los objetivos de venta de satisfacción del cliente y venta personalizada.

Frente a otros métodos de priorización ágil, con nuestro método propuesto se hace explícita la relación entre cada requisito software del producto y el valor del stakeholder implicado. De la misma manera, los responsables del desarrollo pueden priorizar aquellos objetivos que crean más relevantes a pesar de los resultados (como en cualquier técnica ágil), o incluso seleccionar solo aquellos que favorezcan a un stakeholder en concreto, dado que la trazabilidad entre las actividades y los elementos intencionales de GRL así lo permiten.

## **5. Conclusiones**

En esta sección se llevará a cabo una valoración de los resultados obtenidos por este trabajo, se valorará el conjunto de este y por último se contemplarán qué trabajos se han considerado para ser realizados en el futuro.

### **5.1 Resultados**

En este trabajo se ha iniciado la propuesta un método para la especificación del valor del cliente utilizando GRL para su modelado. En primer lugar, se ha realizado una revisión acerca del concepto de valor y en consecuencia se ha elegido un medio de representación capaz de reflejar tanto el carácter económico o de negocio del valor como elementos menos tangibles como la satisfacción de los usuarios de un producto software. Se ha elegido a GRL como lenguaje para el modelado del valor y se ha propuesto un proceso que guíe la construcción de un modelo de valor en este lenguaje. A partir de este modelo de valor, se ha propuesto su integración en un proceso de desarrollo que combine el uso del modelado del valor con las metodologías ágiles, aprovechando conceptos ágiles como la participación del cliente en el proceso, el desarrollo iterativo y centrado en la entrega temprana de valor y el concepto de *backlog*. Se ha integrado el modelo de valor en el proceso formando parte de la fase temprana de la ingeniería de requisitos, permitiendo la derivación de requisitos a partir de los elementos que son de valor para el cliente. Y también se ha propuesto el uso del modelo de valor como elemento para la priorización de los requisitos del software y se ha aportado un medio para cuantificar este valor e integrarlo en un proceso de desarrollo junto a otras prácticas ágiles de estimación de coste y priorización.

Por último, se ha iniciado la exploración de la transformación de modelos de objetivos GRL en los que se representa el valor en modelos de proceso de negocio en BPMN y se han aportado algunas guías para realizar la transformación del primero en el segundo. Para comprobar la consistencia de la propuesta se ha puesto en práctica en una prueba de concepto.

La principal ventaja detectada del proceso propuesto es que pone en primer plano el valor del cliente, siendo el elemento crucial para detectar las necesidades de este y en base a estas necesidades representadas en un modelo de valor, derivar las características del sistema. De esta manera, se tiene la certeza de saber a qué aspectos de valor contribuyen cada uno de los procesos y elemento software diseñados, de forma que se pueda trazar, una vez construido el software, el rendimiento y calidad de este con las aportaciones al valor del cliente.

También, el modelado del valor permite al cliente expresarse en términos de su dominio de manera que se mantenga un nivel de abstracción alto dónde el cliente se siente cómodo para expresar sus necesidades y dónde sus comentarios son de mayor utilidad para el equipo de desarrollo.

El modelado del valor a su vez, es independiente de la aplicación a desarrollar o de los sistemas de información con lo que puede ser reutilizada y ser una herramienta de análisis estratégico desde el punto de vista del negocio. Así pues, puede utilizarse un mismo modelo de valor, para diseñar distintas aplicaciones o explorar nuevos escenarios.

El método propuesto es ligero y está basado en prácticas comunes de la industria a la vez que es adaptable a las prácticas concretas de la organización pudiendo usarse como una práctica ágil más e incorporarse al flujo de trabajo de un equipo de desarrollo de software sin grandes costes.

También se han encontrado puntos de mejora. A pesar de que el modelado del valor en GRL está soportado por jUCMNav, una herramienta de gestión de modelos basada en eclipse, esta no es completamente satisfactoria para el modelado. En primer lugar, el diseñador no es demasiado usable y en segundo, se echan de menos funciones que permitan ver solamente ciertas perspectivas o seleccionar solamente subconjuntos de elementos relevantes, de manera que se mejore la legibilidad de los modelos.

Los modelos en GRL por su complejidad gráfica, no escalan bien, causando que para sistemas complejos sea necesario realizar herramientas que permitan vistas desde distintas perspectivas (por ejemplo, centrándose solamente en los elementos relevantes para un actor). Así se conseguiría reducir la complejidad gráfica, aunque también puede optarse por otro tipo de representaciones más sencillas por ejemplo en formato de tabla u otras.

Por último, otro inconveniente encontrado es que a pesar de que los modelos de valor expresados en GRL son en esencia sencillos, (no poseen muchos elementos gráficos diferentes y su significado general es intuitivo) puede resultar difícil comunicar al cliente de manera satisfactoria la información que aparece en ellos, una vez más, por la complejidad que pueden alcanzar cuando se modelan muchos elementos.

## **5.2 Valoración del trabajo**

El trabajo se ha realizado como parte del máster Ingeniería y Tecnología de Sistemas Software. Este trabajo tiene relación directa con los contenidos del máster siendo las asignaturas principalmente relacionadas Ingeniería del Software Avanzada, Ingeniería del Software Experimental, Experiencias en Gestión de Modelos y Modelado de Proceso de Negocio y Organizacional.

El trabajo ha constado de 700 horas de trabajo aproximadas en las que se ha explorado el estado del arte, se han realizado exploraciones de lenguajes candidatos para la representación del valor siendo, e<sup>3</sup>-value la primera candidata, más tarde explorando i\* y más tarde decantándose por GRL; la creación del método y su refinamiento, reuniones de trabajo y finalmente la ejecución de la memoria de TFM.

## **5.3 Trabajo futuro**

Este trabajo es una propuesta inicial para el uso del valor del para dirigir el desarrollo de servicios cloud iterativo e incremental y formará parte del futuro trabajo de doctorado, profundizando en los temas aquí descritos.

Con respecto a la metodología propuesta se va a realizar un experimento para comprobar la validez de la propuesta y realizar mejoras en el caso de ser necesarias.

De igual manera, la transformación de un modelo de valor GRL en un modelo de proceso de negocio necesita de un estudio más detallado para construir unas guías más efectivas, deseablemente permitiendo la automatización del proceso, dado que aquí solo se realiza una primera aproximación al problema. En esta línea, deberá detallarse la definición del



resto de tareas del proceso, en concreto la transformación de procesos de negocio en casos de uso. Revisando la literatura existente, se debería construir una guía que explicite la transformación y, deseablemente, permita la automatización del proceso.

En relación a las mejoras al método, sería relevante estudiar la forma de incorporar al modelo de proceso de negocio los objetivos soft u otros objetivos que representen propiedades no funcionales de manera explícita de forma que estas consideraciones lleguen hasta los desarrolladores que tienden a trabajar en base a requisitos funcionales.

También, deben completarse el soporte al método con herramientas adecuadas a las distintas fases de este para que su puesta en práctica en producción sea factible.

La elaboración de un modelo de valor puede ser un proceso costoso pero dado que muchas organizaciones llevan a cabo procesos de negocio similares, sería posible que la elaboración de un catálogo de modelos de valor, organizados por dominio, contribuyera a simplificar este proceso de manera que el analista solo tuviera que personalizar un modelo del catálogo para ajustarlo a la organización cliente, aprovechándose así del conocimiento previo del área.

También por los mismos motivos, es útil la incorporación de la base de conocimiento de un *framework* como Service Measurement Index (SMI) a la hora de modelar propiedades de calidad de los sistemas software. De esta manera, aparte de contar con un conjunto de propiedades bien definidas con métricas asociadas, se puede aprovechar el conocimiento sobre los impactos de una propiedad en el resto. Ya que estas relaciones influyen en las decisiones de diseño y repercutirán posteriormente en la calidad del sistema.

Como se ha indicado en el contexto, este proyecto está asociado al proyecto Value@Cloud, por lo que es necesario que se integre dentro del flujo de trabajo de este proyecto. De esta manera, debe estudiarse la integración con el Paquete 3 (definición de una arquitectura para el despliegue incremental de servicios cloud). En este paquete tiene por objetivo definir una arquitectura de despliegue de servicios cloud iterativos, de reconfiguración dinámica y autoadaptativos. La integración con este paquete permitirá que las distintas iteraciones, definidas por este método, una vez completadas se desplieguen en la nube de manera incremental, permitiendo que los servicios existentes se adapten de forma automática a la aparición de los nuevos, realizando los cambios a la arquitectura que fueran necesarios de manera también automática.

Este TFM también está relacionado con el Paquete 4 (Definición y gestión de mecanismos de monitorización del valor en tiempo de ejecución), centrado en la monitorización de la calidad y el rendimiento de los servicios desplegados en la nube por este proceso. Este paquete tiene por objetivo monitorizar el valor especificado por el modelo de valor definido en este trabajo, es decir, que el proceso de monitorización obtenga resultados del calidad de los servicios y dichos resultados sirvan para retroalimentar el proceso de desarrollo de nuevos requisitos o se utilicen de entrada para los procesos de reconfiguración dinámica permitidos por el Paquete 3.

## Referencias

- [1] Boehm, B. (1984). *Software Engineering Economics*. IEEE Transactions on Software Engineering, 10(1), 4–21.
- [2] Mishra, J y Mohanty, A. The Software Crisis. *Software Engineering*. Pearson Education, 2011. P. 11-12.
- [3] Biffel, S., Aurum, A., Boehm, B. (2006). *Value-Based Software Engineering*. Value-Based Software Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg. <http://doi.org/10.1007/3-540-29263-2>
- [4] Beck, K., Grenning, J., Martin, R., Beedle, M., Highsmith, J., Mellor, S., Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Cunningham, W., Kern, J., Thomas, D., Fowler, M., Marick, B. (2001). *Manifesto for Agile Software Development*. [www.agilemanifesto.org](http://www.agilemanifesto.org). Agile Alliance.
- [5] Sutherland, J., Schwaber, K. (1995). *Business object design and implementation: OOPSLA '95 workshop proceedings*. The University of Michigan. P. 118. ISBN 3-540-76096-2.
- [6] Beck, K., Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (2nd Edition). Addison-Wesley Professional.
- [7] Racheva, Z., Daneva, M., Sikkil, K., & Herrmann, A. (2010). *Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study*. <http://doi.org/10.1109/RE.2010.27>
- [8] Yu, E. S. K. (1997). *Towards modelling and reasoning support for early-phase requirements engineering*. Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering, P. 226–235. <http://doi.org/10.1109/ISRE.1997.566873>
- [9] Yu, E. (1995). *Modelling Strategic Relationships for Process Reengineering*, Ph.D. thesis, also Tech. Report DKBS-TR- 94-6, Dept. of Computer Science, University of Toronto.
- [10] Gordijn, J. (2003). *Value Based Requirements Engineering: Exploring Innovative e-Commerce Ideas*.
- [11] De Castro, V., Marcos, E., Vara, J. M. (2011). *Applying CIM-to-PIM model transformations for the service-oriented development of information systems*. Information and Software Technology, 53(1), 87–105. <http://doi.org/10.1016/j.infsof.2010.09.002> ,
- [12] Valeria de Castro, M. (2007). *Aproximación MDA para el desarrollo orientado a servicios de sistemas de información web: del modelo de negocio al modelo de composición de servicios web*. <http://www.cchhttp/hdl.handle.net/10115/3335> .
- [13] Guzman, M. (2015). *Especificación de requisitos para servicios cloud dirigido por valor*. <https://riunet.upv.es/handle/10251/55638> .
- [14] Karlsson, J., Ryan, K. (1997). *A Cost-Value-Approach for Prioritizing Requirements*. IEEE Software, 14(5), 67 – 74. <http://doi.org/10.1109/52.605933>
- [15] Azar, J., Smith, R. K., Cordes, D. (2007). *Value-oriented requirements prioritization in a small development organization*. IEEE Software, 24(1), 32–37. <http://doi.org/10.1109/MS.2007.30>
- [16] Rauf, A., AlGhafees, M. (2015). *Gap Analysis between State of Practice and State of Art Practices in Agile Software Development*. 2015 Agile Conference, 102–106. <http://doi.org/10.1109/Agile.2015.21>
- [17] Beck, K., Grenning, J., Martin, R., Beedle, M., Highsmith, J., Mellor, S., Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Cunningham,

- W., Kern, J., Thomas, D., Fowler, M., Marick, B. (2001). *Principles behind the agile manifesto*. <http://agilemanifesto.org/iso/en/principles>
- [18] Wikipedia. *Iterative and incremental development*. [https://en.wikipedia.org/wiki/Iterative\\_and\\_incremental\\_development](https://en.wikipedia.org/wiki/Iterative_and_incremental_development)
- [19] Wikipedia. *Iterative design*. [https://en.wikipedia.org/wiki/Iterative\\_design](https://en.wikipedia.org/wiki/Iterative_design)
- [20] Atlassian. *The product backlog: your ultimate to-do list* <https://www.atlassian.com/agile/backlogs>
- [21] Scheer, A.-W., Nuttgens, M. (2000) *Aris architecture and reference models for business process management*. In Business Process Management. Springer.
- [22] Object Management Group (OMG). (2008) *Business Process Modeling Notation (BPMN) Specification*, version 1.1 edition.
- [23] ITU-T. (2011). *User Requirements Notation (URN) - Language requirements and framework*, 40. <http://www.itu.int/rec/T-REC-Z.150/en>
- [24] van der Aalst, W. M. P., ter Hofstede, A. H. M. (2005) *Yawl: yet another workflow language*. Information Systems, 30(4):245–275.
- [25] Organization for the Advancement of Structured Information Standards (OASIS). (2007) *Web Services Business Process Execution Language (WSPBEL) Specification*, version 2.0 edition.
- [26] Petri, C.A. (2000) PhD thesis, Institut für Instrumentelle Mathematik, Bonn, 1962. G. Knolmayer, R. Endl, and M. Pfahrer. *Modeling processes and workflows by business rules*. In *Business Process Management, Models, Techniques, and Empirical Studies*, pages 16–29, Springer-Verlag.
- [27] Polancic, G. (2014). *The Popularity of BPMN just keeps rising*. Goodelearning. <http://blog.goodelearning.com/bpmn/popularity-bpmn-rising/>.
- [28] Racheva, Z., Daneva, M., Sikkil, K. (2009). *Value creation by agile projects: Methodology or mystery?* Lecture Notes in Business Information Processing, 32 LNBIP, 141–155. [http://doi.org/10.1007/978-3-642-02152-7\\_12](http://doi.org/10.1007/978-3-642-02152-7_12)
- [29] Barnett, L. (2007) *Agile Projects Must Measure Business Value*. Agile Journal (January 2007).
- [30] Patton, J. (2008). *Ambiguous Business Value Harms Software Products*. IEEE Software 25.
- [31] Pettit, R. (2007). *Business Value Applied: Aligning The Day To Day With Business Imperative*. Agile Journal (January 4, 2007),
- [32] AgileConnection. <http://www.agilejournal.com/content/view/206/33/>.
- [33] Rawsthorne, D. (2004) *Managing the Work in an Agile Project*, <http://www.netobjectives.com/files/resources/downloads/ManagingTheWork.pdf>
- [34] Poole, M. (2007). *Business and IT – A Marriage Made in Heaven?* Agile Journal (October 6, 2007), <http://www.agilejournal.com/content/view/627/76/>
- [35] Highsmith, J. (2013). *Determining Business Value*. <http://jimhighsmith.com/determining-business-value/>
- [36] Kaplan, R. S. (2009). *Conceptual Foundations of the Balanced Scorecard*. *Handbooks of Management Accounting Research*, 3, 1253–1269. [http://doi.org/10.1016/S1751-3243\(07\)03003-9](http://doi.org/10.1016/S1751-3243(07)03003-9)
- [37] Porter, M. E. (1985), *The Competitive Advantage: Creating and Sustaining Superior Performance*. NY: Free Press.
- [38] Drucker, P. (1955). *The Practice of Management*. Harper. Revisado por ed. Butterworth-Heinemann en 2007.

- [39] Mylopoulos, J., Chung L., Nixon B. (1992). *Representing and Using Nonfunctional Requirements: A Process-Oriented Approach*. IEEE Transactions on Software Engineering, vol. 18, no. 6, pp. 483-497.
- [40] Sulaiman, T., Barton, B., Blackburn, T. (2006). *AgileEVM-earned value management in Scrum Projects*. Proceedings of the Agile Conference, Minneapolis (MN), USA, 23-28 July, 2006, 10 pp. – 16. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1667558](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1667558)
- [41] Boehm, B. W. (2006). *Value-based software engineering: Overview and agenda*. Value-Based Software Engineering, 3–14. [http://doi.org/10.1007/3-540-29263-2\\_1](http://doi.org/10.1007/3-540-29263-2_1)
- [42] Saaty, T. (1985) *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. ISBN 0-07-054371-2. McGraw-Hill.
- [43] Karlsson, J., & Ryan, K. (1997). *A Cost-Value-Approach for Prioritizing Requirements*. IEEE Software, 14(5), 67 – 74. <http://doi.org/10.1109/52.605933>
- [44] Highsmitt, J. (2010). *Agile Project Management: Creating Innovative Products* (2º Edición). ISBN 0-321-21977-5, Pearson Education/Addison-Wesley.
- [45] Torrecilla-Salinas, C. J., Sedeño, J., Escalona, M. J., Mejías, M. (2015). *Estimating, planning and managing Agile Web development projects under a value-based perspective*. Information and Software Technology, 61, 124–144. <http://doi.org/10.1016/j.infsof.2015.01.006>
- [46] Wikipedia. *Earned Value Management*. [https://en.wikipedia.org/wiki/Earned\\_value\\_management](https://en.wikipedia.org/wiki/Earned_value_management)
- [47] J.D. McKeen, H. Smith. (2003) *Making IT Happen: Critical Issues in IT Management*, Hoboken, NJ.
- [48] Bleistein, S.J, Cox, K., Verner, J.M., Phalp, K. (2006). *A requirements analysis framework for validating strategic alignment of organizational IT based on strategy, context, and process* <http://www.agilejournal.com/content/view/211/76/>
- [49] Gordijn, J., Yu, E., Raadt, B. Van Der. (2006). *e-Service Design Using i\* and e3-value Modelling*, IEEE Software 23(3).
- [50] OpenOME. OpenOME, an open-source requirements engineering tool. <https://se.cs.toronto.edu/trac/ome/wiki>
- [51] Braun, E., Amyot, D., Lethbridge, T. (2016). *Generating Software Documentation in Use Case Maps from Filtered Execution Traces*. University of Ottawa.
- [52] International Telecommunications Union. About ITU. <https://www.itu.int/en/about/Pages/default.aspx>
- [53] Amyot, D., Mussbacher, G. (2011). *User requirements notation: The first ten years, the next ten years*. Journal of Software, 6(5), 747–768. <http://doi.org/10.4304/jsw.6.5.747-768>
- [54] Object Management Group. (2015). *Value Delivery Metamodel (VDM)*, (October). <http://www.omg.org/cgi-bin/doc?bmi/09-03-09>
- [55] Osterwalder, A. (2004). *The business model ontology: a proposition in a design science approach*. Tesis, Université de Lausanne.
- [56] Biem, A., Caswell, N. (2008). *A Value Network Model for Strategic Analysis*. HICSS.
- [57] McCarthy, E. (1982). *The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment*. The Accounting Review, pp. 554-78.

- [58] Rother, M., Shook, J. (1999) *Learning to See: value-stream mapping to create value and eliminate muda*. Brookline, MA: Lean Enterprise Institute.
- [59] De Man, H. (2013) *VDML: Value Delivery Modeling Language*. <http://www.slideshare.net/hdman/vdml-guest-lecture-ugent-november-2013>
- [60] The NEFFICS Consortium. *Value Delivery Modeling Language (VDML): Notation Example*. [http://practices.modelbased.net/practice.business.value\\_delivery\\_modelling.bas-e-sintef/guidances/examples/vdml\\_notation\\_682B171D.html](http://practices.modelbased.net/practice.business.value_delivery_modelling.bas-e-sintef/guidances/examples/vdml_notation_682B171D.html)
- [61] Svensson, R. B., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., & Feldt, R. (2011). *Prioritization of Quality Requirements: State of Practice in Eleven Companies*. Requirements Engineering Conference, 2011 19<sup>th</sup> IEEE International, 69–78.
- [62] Achimugu, P., Selamat, A., Ibrahim, R., & Naz, M. (2014). *A systematic literature review of software requirements prioritization research*. Information and Software Technology, 56(6), 568–585. <http://doi.org/10.1016/j.infsof.2014.02.001>
- [63] Racheva, Z., Daneva, M., Sikkel, K., & Herrmann, A. (2010). *Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study*. <http://doi.org/10.1109/RE.2010.27>
- [64] OBS Business School. Blog Investigación. Universitat de Barcelona. 12 Técnicas para la estimación de costes en proyectos. <http://www.obs-edu.com/blog-investigacion/project-management/12-tecnicas-para-la-estimacion-de-costes-en-proyectos/>
- [65] Wolf Project. *Cómo estimar los costes de un proyecto*. <http://wolfproject.es/como-estimar-los-costos-de-un-proyecto/>
- [66] Popli, R., Chauhan, N. (2014). *Cost and effort estimation in agile software development*. International Conference on Optimization, Reliability and Information Technology 2014, 57–61. <http://doi.org/10.1109/ICROIT.2014.6798284>
- [67] Usman, M., Mendes, E., Weidt, F., Britto, R. (2014). *Effort estimation in agile software development*. Proceedings of the 10th International Conference on Predictive Models in Software Engineering - PROMISE '14, (September), 82–91. <http://doi.org/10.1145/2639490.2639503>
- [68] Wikipedia. *Planning Poker*. [https://en.wikipedia.org/wiki/Planning\\_poker](https://en.wikipedia.org/wiki/Planning_poker)
- [69] Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledge* (quinta edición). Project Management Institute. ISBN 978-1-935589-67-9
- [70] De Castro, V., Mesa, J. M. V., Herrmann, E., & Marcos, E. (2008). *From real computational independent models to information system models: An MDE approach*. CEUR Workshop Proceedings, 389, 46–60.
- [71] Chen, P. (2007). *Goal-Oriented Business Process Monitoring*. Master Thesis School of computer Science, Carleton University, Ottawa.
- [72] Markovic, I., Kowalkiewicz, M. (2008). *Linking business goals to process models in semantic business process modeling*. Proceedings - 12th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2008, 332–338. <http://doi.org/10.1109/EDOC.2008.43>

- [73] Braubach, L., Pokahr, A., Jander, K., Lamersdorf, W., & Burmeister, B. (2010). *Go4Flex: Goal-oriented process modelling*. *Studies in Computational Intelligence*, 315, 77–87. [http://doi.org/10.1007/978-3-642-15211-5\\_9](http://doi.org/10.1007/978-3-642-15211-5_9)
- [74] Decreus, K., Snoeck, M., Poels, G. (2009). *Practical challenges for methods transforming i\* goal models into business process models*. *Proceedings of the IEEE International Conference on Requirements Engineering*, 15–23. <http://doi.org/10.1109/RE.2009.25>
- [75] Sguran, M., Hilbert, C., Frankova, G. (2008). *Secure workflow development from early requirements analysis*. *Proceedings of the 6th IEEE European Conference on Web Services, ECOWS'08*, 125–134. <http://doi.org/10.1109/ECOWS.2008.13>
- [76] Pistore, M., Roveri, M., & Busetta, P. (2004). *Requirements-driven verification of Web services*. *Electronic Notes in Theoretical Computer Science*, 105(1–4 SUPPL.), 95–108. <http://doi.org/10.1016/j.entcs.2004.05.005>
- [77] Koliadis, G., Vranesevic, A., Bhuiyan, M., Krishna, A., Ghose, A. (2006). *Combining i\* and BPMN for Business Process Model Lifecycle Management*. *BPM 2006 Workshops, LNCS 4103*, 416–427. [http://doi.org/10.1007/11837862\\_39](http://doi.org/10.1007/11837862_39)
- [78] Lo, A., & Yu, E. (2007). *From Business Models to Service-Oriented Design: A Reference Catalog Approach*. *26th International Conference on Conceptual Modeling*, 4801, 87–101. [http://doi.org/10.1007/978-3-540-75563-0\\_8](http://doi.org/10.1007/978-3-540-75563-0_8)