Facilitando la Asignación de Prioridades a los Requisitos

Graciela Hadad[(1)] Jorge Doorn[(1)(2)] Marcela Ridao[(2)] Gladys Kaplan[(1)]
gracielahadad@gmail.com *jdoorn@exa.unicen.edu.ar* *mridao@exa.unicen.edu.ar* *gladyskaplan@gmail.com*

(1) Departamento de Ingeniería e Investigaciones. Tecnológicas, UNLaM, Argentina
(2) INTIA, Facultad de Ciencias Exactas, UNICEN, Argentina

Abstract

Existe una gran variedad de propuestas y estándares internacionales que ofrecen guías de contenido y estilo para la generación de un documento de especificación de requisitos (SRS). Adicionalmente, la literatura da cuenta de diversos métodos para asignar prioridades a los requisitos, que parten de un conjunto de requisitos perfectamente individualizados, frecuentemente plasmados en un SRS. En este artículo se presenta una estrategia alternativa para asignar prioridades a los requisitos, que se apoya en el proceso previo que les dio origen: su derivación desde escenarios y la descomposición de objetivos en sub-objetivos.

1. Introducción

Estrategias en la ingeniería de requisitos basadas en el uso de modelos escritos en lenguaje natural han intentado contribuir a la mejora del proceso de desarrollo de software, atendiendo principalmente aspectos socio-culturales de las organizaciones, motivando la comunicación y la colaboración continua entre los involucrados. Uno de estos enfoques es el propuesto por Leite et al. [2], que se sustenta en el uso de dos modelos en lenguaje natural: el léxico extendido del lenguaje (glosario con términos empleados en el universo de discurso) y los escenarios. Estos últimos permiten describir situaciones presentes en el universo de discurso, facilitando la adquisición de conocimiento sobre la realidad actual y sus problemas, como así también, permiten idear situaciones deseadas cuando un nuevo sistema de software o una adaptación a sistemas existentes entre en vigencia en el universo de discurso. A este segundo conjunto de escenarios, que representan esas situaciones esperadas en el futuro se los denomina efectivamente Escenarios Futuros (EF) [1]. En ellos están inmersos los requisitos del sistema de software de una manera, a veces, no tan visible o fácilmente detectable, y donde muchas veces un requisito se encuentra presente en más de un escenario o una oración dentro de un escenario involucra dos o más requisitos de diferente naturaleza. Es por ello que se ha propuesto una estrategia que evidencia en forma precisa los requisitos que se hallan empotrados en los EF [9].

Dada la necesidad de establecer prioridades a los requisitos, ya que no es viable en proyectos de mediana a gran envergadura implementar todos los requisitos de una sola vez, se ha propuesto un abanico de posibilidades para encarar esta actividad [3] [13] [18] [19] [20] [21] [26]. Algunos métodos establecen cálculos minuciosos para asignar prioridades según una diversidad de atributos anexos; otros consideran múltiples puntos de vista y sus intereses; mientras otros tienen en cuenta la dependencia existente entre requisitos. Estos dos últimos aspectos hacen a lo que podría denominarse compatibilizar prioridades.

En general, las referidas propuestas arrancan de un conjunto de requisitos, contenido en un documento de definición de requisitos o bien en fichas de especificación basadas en templates, y asignan prioridades sin valerse del proceso que dio origen a dicho conjunto.

En el presente artículo se muestra que la asignación de prioridades a requisitos es una actividad que se realiza en forma muy diferente, y más simple, cuando los requisitos se extraen de un modelo coherente del proceso del negocio, como son los EF.

La sección siguiente incluye una breve descripción de varios métodos que tratan la asignación de prioridades a los requisitos, junto con posibles estructuras para especificar requisitos y sus atributos. En la sección 3 se muestra el meta-modelo de especificación de requisito que luego se utilizará en la sección 4, donde se presenta resumidamente el proceso de extracción de requisitos de un conjunto de EF. En la sección 5 se detalla una estrategia para la asignación de prioridades a los requisitos basándose en el proceso mismo de definición de requisitos. En la sección 6 se presenta la aplicación de la estrategia de asignación de prioridades a un caso de estudio. Finalmente, en la sección 7 se exponen conclusiones y se proponen trabajos futuros.

2. Especificación de requisitos del software

No existe un acuerdo aceptado sobre cuáles son los atributos intrínsecos de los requisitos que permitan su mejor representación y gestión, tal es la diversidad expuesta en [4] [5] [6] [7] [8] [9] [10] [11] [12] [14] [15]. Uno de dichos atributos, la prioridad, es además objeto de un tratamiento especial, a partir del cual se suele establecer la implementación, postergación o, hasta incluso, cancelación de un requisito [30] [34] [36]. Es frecuente observar que para el cálculo de este atributo se utilizan otros atributos tales como la factibilidad, criticidad, beneficios, costos, entre otros [25] [26] [27].

2.1. Prioridades de Requisitos

La tarea de asignar prioridades requiere de la participación de clientes y usuarios con cierto nivel de decisión y puede realizarse de diversas maneras, tales como reuniones, cuestionarios y otras. Se debe determinar la importancia relativa que tiene un requisito para los clientes y usuarios, y organizar aquellos requisitos que deben implementarse inicialmente frente a aquellos que pueden postergarse.

Al asignar prioridades, se deben tener en cuenta la dependencia entre requisitos, la multiplicidad de intereses de los clientes y usuarios, las limitaciones de recursos, las necesidades del negocio, las imposiciones del mercado y los costos de implementación, entre otros factores. Por ende, los ingenieros de software deberán orientar a los clientes y usuarios respecto a estas contingencias.

La asignación de prioridades ha sido muy estudiada por una variedad de autores que han definido estrategias simples o complejas, que van desde una asignación que califica al requisito como importante o no, como obligatorio o postergable (suspendido) hasta un ranking de importancia de los requisitos. Dentro de estas técnicas de asignación de prioridades a requisitos se encuentran: Asignación Numérica [16] que divide los requisitos en tres grupos: obligatorios, deseables y no esenciales; Ranking se basa en una escala ordinal donde al requisito más importante se le da el valor 1 y al menos importante el valor N siendo N la cantidad de requisitos, y la lista de ranking se obtiene aplicando por ejemplo el método de ordenamiento burbujeo o árbol binario de búsqueda (ver en [3]); AHP (Analytic Hierarchy Process) [17] [18] basada en la técnica de "pairwise comparison" estimando un valor relativo de importancia de los requisitos; QFD (Quality Function Deployment) [19] basada en la asignación numérica de prioridades respecto a una escala absoluta; Método de 100 Puntos [20] donde a cada participante se le dan 100 puntos para aplicar votando a favor de los requisitos más importantes para él; Teoría W (también denominada Win-Win) [21] [22] donde cada involucrado da un ranking privadamente a los requisitos considerando aquellos que está dispuesto a abandonar y luego se realiza la negociación; Requirements Triage [6] incluye dar prioridades relativas a los requisitos, estimar los recursos necesarios para satisfacerlos y seleccionar un subconjunto de requisitos para optimizar la probabilidad de éxito; Juego de Planeamiento [23] usado en Programación Extrema sobre las historias de usuarios donde se agrupan los requisitos en grupos y luego se dan rankings dentro de cada grupo; y SERUM [24] usa estimaciones de costo, beneficio, riesgo de desarrollo y reducción de riesgo operacional para dar prioridades.

Otro aspecto a considerar adicional a asignar prioridades, es la compatibilización de las mismas debido a la dependencia entre requisitos. En este caso también se han propuesto varias técnicas, como por ejemplo: el método de Wiegers [25] se basa en calcular una prioridad para cada requisito en base a cuatro atributos: beneficio para el cliente, penalidad si el requisito no se incluye, costo de implementación y riesgo técnico, asignando a cada atributo un valor entre 1 y 9, antes de calcular prioridades se determina que todos los requisitos tengan el mismo nivel de abstracción y se considera si hay requisitos vinculados para incluir sólo el requisito dominante; EVOLVE [26] basada en un algoritmo genético que considera distintos puntos de vista de los involucrados, restricciones de esfuerzo, restricciones de riesgo y dependencias entre requisitos; Win-Win Cuantitativo [27] que a diferencia de la técnica Win-Win de Boehm incluye métodos cuantitativos pues usa AHP para determinar las preferencias de los involucrados, y estos resultados se combinan para evaluar la factibilidad de subconjuntos de requisitos alternativos considerando sus esfuerzos de implementación relativos; y Requirements Prioritization Framework [13] [28] donde los involucrados dan una tasa de importancia a los requisitos y a los objetivos del negocio, luego se buscan las dependencias de los requisitos para dar prioridades más eficientemente y finalmente se aplican técnicas de análisis de riesgo para descubrir valoraciones subjetivas o corporativas de los involucrados.

2.2. Atributos de los Requisitos

Los requisitos del sistema de software se plasman en documentos, como el Concepto de Operaciones [8] y la Especificación de Requisitos de Software [10], o bien en plantillas que incluyen ciertos atributos [11]

[34] [30] [35] [36] [15], los que pueden variar de un proyecto a otro, y que muchas veces se ven limitados por la herramienta adoptada para la administración o el modelado de los requisitos. Los atributos asociados a los requisitos le darán contexto a cada requisito, posibilitando su identificación, clasificación o selección, y permitiendo el control mismo del proceso de definición de requisitos [37].

Davis en [34] propone, para la gestión de requisitos, incluir los siguientes atributos: beneficio del cliente, esfuerzo de implementación, prioridad de desarrollo, estado, autores (persona responsable de escribir la especificación o persona que identificó la necesidad), parte responsable (persona que asegura la satisfacción del requisito), fundamento, fecha de creación o modificación, versión y relación con otros requisitos.

SWEBOK [35] propone, con fines de gestión e interpretación de los requisitos, incluir como atributos: la clasificación en distintas dimensiones (requisitos funcionales o no funcionales, derivado o impuesto, requisito sobre el producto o el proceso, prioridad, alcance, volatilidad / estabilidad), el método de verificación o plan de prueba de aceptación, el fundamento, el origen del requisito, la historia de cambios y un identificador único del requisito.

Kotonya & Sommerville [30] proponen para especificar requisitos un meta-modelo de base de datos donde incluyen: el identificador del requisito, descripción del requisito, fecha de creación, fecha de modificación, origen del requisito (persona, documento u otro requisito), fundamento, estado, lista de requisitos dependientes, lista de requisitos de los que depende, vínculos a modelos y un comentario.

En [36] se presenta una plantilla con los siguientes atributos: número (identificador del requisito), título (indica la naturaleza del requisito), texto (descripción), detalles y restricciones (características funcionales o dimensiones), fecha de revisión (cuándo fue aceptado), número de revisión de la versión actual y criticidad (obligatorio, deseado u opcional).

3. El Meta-Modelo de Especificación de Requisito

Para describir en detalle cada requisito se propone un meta-modelo de Especificación de Requisito (ver Figura 1), que tiene en consideración exclusivamente atributos intrínsecos al requisito. Es decir, se han excluido de él atributos relacionados con el versionado, la rastreabilidad y la interdependencia de requisitos.

Este meta-modelo puede adaptarse a las necesidades de cada proyecto en particular, dado que la propuesta presenta atributos básicos: una descripción, un tipo configurable para brindarle contexto al requisito, un fundamento que aclare su propósito o el origen de su existencia, un estado que indique la situación en la que se encuentra el requisito respecto al proceso de desarrollo, y un conjunto de atributos extras que pueden ser necesarios para establecer la prioridad del mismo, tales como volatilidad, criticidad, factibilidad, riesgo y costo de implementación.

Requisito: descripción del requisito.

Tipo: indica si se trata de un RF o un RNF, o el tipo de RNF al que corresponde, o el tipo según alguna otra clasificación utilizada.

Fundamento: razón de la existencia del requisito.

Estado: condición bajo la cual se encuentra el requisito, puede ser por ejemplo propuesto, aceptado, implementado, validado, etc.

Volatilidad: grado de estabilidad del requisito.

Prioridad: importancia relativa que tiene el requisito para los clientes y usuarios.

Criticidad: necesidad relativa de implementación, puede indicarse si es obligatorio, deseable u opcional, o mediante un ranking de necesidad.

Factibilidad: posibilidad de implementación en el proceso del negocio, ya sea por razones sociales, tecnológicas, ambientales, económicas, etc.

Riesgo: calificación en función de las consecuencias en el proceso del negocio por su implementación.

Costo de implementación: esfuerzo para implementar el requisito en el sistema de software.

Figura 1. Meta-Modelo de Especificación de Requisito

Cabe aclarar que, respecto a las dependencias entre requisitos, ellas quedan claramente establecidas en el momento de extraer los requisitos del conjunto de EF y serán específicamente consideradas en la asignación de prioridades.

4. El proceso de explicitar los requisitos

Este proceso, presentado inicialmente en [9] y detallado en [40], tiene por objetivo individualizar los requisitos del software que se hallan, en su gran mayoría, en los EF construidos en una etapa previa. Los EF son contenedores tanto de requisitos funcionales (RF) como de requisitos no funcionales (RNF) y son la entrada principal a este proceso.

Las actividades involucradas en el proceso, que se presenta mediante un modelo SADT en la Figura 2, son:

1. Generar la lista de requisitos,
2. Dar atributos a los requisitos,
3. Organizar los requisitos en un SRS, y
4. Verificar el SRS.

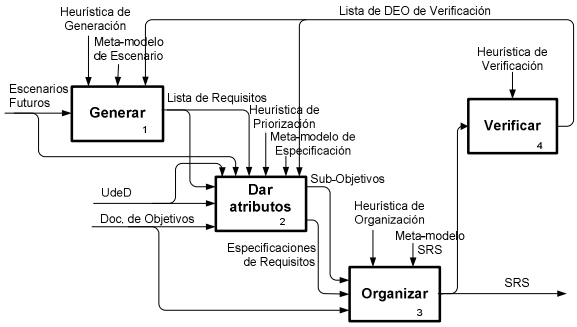

Figura 2. SADT del proceso de explicitar Requisitos de Software

La primera actividad se encarga de buscar requisitos en los EF, armando una lista de requisitos. La heurística de generación [9] indica entonces cómo se extraen los requisitos empotrados en los EF y básicamente expresa que:

a) RF se pueden derivar de:
   - ✓ episodios donde participa como actor el sistema de software;
   - ✓ tratamiento de excepciones donde participa como actor el sistema de software;
   - ✓ causa de excepciones;
   - ✓ condiciones de episodios donde en el episodio participa como actor el sistema de software.

b) RNF se pueden derivar de:
   - ✓ restricciones donde en el episodio participa como actor el sistema de software;
   - ✓ causa de excepciones;
   - ✓ información asociada a recursos que proviene del léxico extendido del lenguaje.

La Figura 4 muestra un ejemplo de lista de requisitos extraída de un solo escenario futuro, donde se aprecian 8 RF.

La lista de requisitos se utiliza en el paso siguiente sin requerir previamente ninguna actividad de análisis, pues los EF utilizados ya pasaron por procesos de inspección [38], de validación y de negociación [9]. Durante esta actividad, se elicitan y acuerdan los atributos de cada requisito para completar su especificación. El atributo más relevante que se establece es la prioridad del requisito. Para ello se aplica la heurística de priorización, que se detalla en la sección 5, donde se utiliza como entrada el documento de objetivos del sistema junto con nueva información capturada del UdeD.

En la tercera actividad se organizan los requisitos según las características del proyecto y se redacta un SRS con la información contenida en las Especificaciones de Requisitos, siguiendo la heurística de organización y en base a algún estándar que se utilice o adopte para el proyecto en curso.

Este documento pasa por un proceso de verificación para comprobar su correcta generación, asegurando que la extracción de requisitos desde los EF sea exhaustiva y la asignación de prioridades sea consistente.

En resumen, a partir de los EF, el documento de objetivos e información adicional elicitada del UdeD, se genera un documento SRS como producto final de este proceso.

En el presente artículo se detallará la segunda actividad de este proceso, en particular lo que respecta a los atributos asociados con el concepto de prioridad.

5. Estrategia para Asignar Prioridades

El fundamento del requisito se establece teniendo en cuenta su origen, es decir, su participación en uno o más EF ayudando a cumplir un objetivo.

Para el resto de los atributos presentados en la Figura 1, y probablemente para otros atributos que se deseen agregar, es necesario definir una escala de valores admisibles. Estas escalas podrán tener unos pocos niveles cualitativos o podrán ser escalas numéricas arbitrarias. En particular, de conservarse el atributo relacionado con el costo de implementación del requisito, es preferible que el mismo se exprese en forma cualitativa o en una escala numérica con pocos valores.

En cuanto al estado, en principio se debe partir de un estado tal como "propuesto", pero una vez que se le hayan asignado todos sus atributos se modificará a un estado "aceptado" o similar. Actualizaciones al estado irán surgiendo durante el resto del proceso de desarrollo del software.

Es difícil definir la volatilidad de los requisitos en una etapa tan temprana del proceso de desarrollo del software. Es cierto que de algunos requisitos se puede conocer su condición de inestable por el contexto en que el mismo fue creado, en virtud de muchos indicadores tales como la dificultad para definir con precisión todo o parte del escenario futuro que le dio origen. Desafortunadamente los requisitos con aparente baja volatilidad pueden ser muy afectados por avatares del proceso del negocio o del contexto en el que el mismo se desenvuelve.

De todos los atributos restantes: prioridad,

criticidad, riesgo, factibilidad y costo de implementación, el que desencadena decisiones acerca de cómo seguir en el proceso es la prioridad. Los otros atributos pueden influir o ayudar en la asignación de la prioridad.

Cabe aclarar que al haberse obtenido la lista de requisitos de un conjunto de EF ya validados y negociados con los clientes y usuarios, la tarea de asignación de prioridades se debe realizar con una visión macroscópica. Es decir, se aspira a asignar prioridades a conjuntos consistentes de requisitos y no a requisitos individuales.

En este punto es importante reexaminar el propósito de asignar prioridades a los requisitos. Este atributo es necesario para resolver la implantación de los requisitos en el futuro sistema de software y eventualmente la división del proyecto en etapas donde los requisitos de mayor prioridad serían materializados en las primeras etapas. Esta visión del posible uso de la prioridad pone en conflicto todo el proceso de software en que la elicitación de requisitos se halla inmersa. Se ha adoptado una modalidad atómica para la obtención de requisitos y repentinamente se cambia a un proceso incremental y/o iterativo. En otros términos, en un proceso en cascada o similar se implementarían todos los requisitos elicitados en forma monolítica, sin puestas en servicio parciales, por lo que la asignación de prioridad no es una actividad necesaria. El aparente o real conflicto entre la elicitación de requisitos en forma total y su implementación por etapas no es necesariamente un problema, ya que puede provenir de una decisión tomada en forma explícita. Si este fuera el caso, la asignación de prioridades se realimenta sobre los EF, los cuales deben ser reestudiados para evaluar en su contexto cada implementación parcial. Si se hubiera seguido estrictamente un proceso iterativo y/o incremental, tampoco sería necesaria la asignación de prioridad, ya que la misma fue asignada al definir el alcance de la iteración.

En el caso en que se trate de una situación en la que se ha definido una única lista de requisitos a partir de la cual se seleccionará un subconjunto que será materializado en una implantación parcial, es necesario precisar la estrategia con la que se definirán las prioridades.

Como primera actividad para el establecimiento de prioridades, se debe seleccionar la técnica de asignación de las mismas. Independientemente de la técnica elegida, se propone a continuación un mecanismo nuevo para la compatibilización de las prioridades, en virtud de que en este proceso la lista de requisitos ha sido obtenida de un modelo del proceso del negocio coherente. Estrictamente se propone a continuación un mecanismo que asigne las prioridades en forma consistente desde el primer momento, haciendo innecesaria toda actividad posterior de compatibilización.

Esta propuesta se basa en asignar prioridades a conjuntos de requisitos que satisfacen algunos de los objetivos o sub-objetivos del sistema de software. Es importante enfatizar que ya durante la generación de los EF fue necesario establecer con claridad el objetivo del sistema de software [2] para seleccionar la estrategia de construcción.

Diversos autores [29] [31] [32] [33] [39] han propuesto, con otro propósito, abstracciones y refinamientos de objetivos generales u objetivos estratégicos para encontrar sub-objetivos o super-objetivos según sea el caso. En dichas propuestas se ha descripto en detalle un proceso que identifica objetivos del negocio y construye una jerarquía de objetivos y sub-objetivos. En este artículo se adopta el uso de la misma estrategia pero sobre los objetivos del sistema de software. Las abstracciones fueron realizadas durante la definición del objetivo general del sistema, antes de construir los EF [2], mientras que los refinamientos deben realizarse en este punto, al asignar prioridades a los requisitos.

Luego, la tarea de asignar prioridades a los requisitos consiste en (heurística de priorización):

i. Dividir el objetivo general del sistema en sub-objetivos, utilizando las técnicas orientadas a objetivos mencionadas.
ii. Continuar la división de los sub-objetivos en sub-objetivos más detallados hasta que se pueda asegurar que cada una de las hojas del árbol admita una única prioridad sin incoherencia interna. En general, es suficiente con uno o dos niveles de descomposición.
iii. Asignar prioridades a las hojas del árbol con los clientes y usuarios.
iv. Relacionar los EF y los sub-objetivos, considerando qué EF satisfacen cada sub-objetivo. No se consideran los EF donde no interviene el sistema de software bajo estudio. Puede ocurrir que un escenario futuro participe en el cumplimiento de más de un sub-objetivo y que un sub-objetivo sea satisfecho por más de un escenario futuro.
v. Para todos los EF que materializan un solo sub-objetivo o más de un sub-objetivo con la misma prioridad, trasladar la prioridad del o de los sub-objetivos a los requisitos que dio lugar el escenario futuro.
vi. Para todos los EF que participan en dos o más sub-objetivos, con al menos dos prioridades diferentes, distinguir qué requisitos del escenario futuro están involucrados con qué sub-objetivo y asignarle su prioridad. Si el requisito satisface dos o más sub-

objetivos con distinta prioridad entonces se le asigna la prioridad más alta.

vii. Para RNF no vinculados a EF, asociar, de ser posible, a sub-objetivos y trasladar su prioridad. Si se asocia a más de un sub-objetivo con distinta prioridad, asignarle la prioridad más alta.

viii. Si un RNF no puede asociarse a uno o más sub-objetivos en particular, esto implica que está relacionado con todos los sub-objetivos y, por lo tanto, se requerirá intervención de los clientes y usuarios.

ix. Cuando un escenario futuro participa en dos o más sub-objetivos con al menos dos prioridades diferentes, es necesario rearmar el escenario futuro en uno o más escenarios de transición, que describan la parte correspondiente al proceso del negocio que existirá en la iteración correspondiente.

En resumen, siguiendo esta heurística, se asignan prioridades a sub-objetivos, las que se trasladan a los requisitos involucrados en la satisfacción de los mismos. De esta manera se facilita la asignación de prioridades a requisitos individuales, pues el consenso a lograrse sobre las prioridades se basa en un conjunto numéricamente inferior de sub-objetivos.

Cabe aclarar que, aunque en las pautas indicadas arriba se hace mención sólo al atributo prioridad, se debe tener en cuenta que dependiendo del criterio de cálculo de la prioridad, este atributo podrá estar representando la importancia o beneficio del requisito para los clientes, o el impacto de los otros atributos, como ser criticidad, factibilidad, costo de implementación y riesgo. La dependencia entre la prioridad y estos otros atributos no puede calcularse individualmente sobre cada requisito sino que debe hacerse en el conjunto de requisitos relacionados por el sub-objetivo. Este aspecto abre una nueva problemática que puede antagonizar con los cálculos de prioridad referenciados en la sección 2.1, pero que excede el alcance del presente artículo.

6. Caso de estudio

Con el fin de analizar la aplicación de la estrategia propuesta, se han evaludado varios casos de estudio. Estos casos habían sido completamente desarrollados con anterioridad, incluyendo la lista de requisitos, sin priorizar. Pudo observarse que en todos ellos era posible efectuar la descomposición del objetivo general del sistema en sub-objetivos, para luego asignar prioridades a los requisitos. En muchos de los casos analizados, la aplicación de la estrategia implicaba la necesidad de construir escenarios de transición a partir de escenarios comunes a más de un sub-objetivo. Cada uno de estos nuevos escenarios hubiera quedado conformado por algunos componentes del escenario futuro original, y otros elementos del universo de discurso actual, correspondientes a la porción del escenario asociado al sub-objetivo de menor prioridad.

Entre los casos analizados, se decidió presentar en este trabajo el caso de estudio de Producción y Acopio de papas de alta calidad. Este caso describe la relación entre productores de papa y una empresa acopiadora. La selección de este caso se debe a la sencillez y claridad de su estructura organizacional, lo cual permite presentar la aplicación de la estrategia de una manera simple y clara. En la Tabla 1, se presenta la lista de EF correspondientes a este caso de estudio.

Tabla 1. Escenarios Futuros correspondientes al caso de estudio

Chequear las variables de la producción
Iniciar la operación de canje de semilla de papa
Iniciar la operación de venta de papa
Pedir estadísticas
Realizar consultas actuales
Realizar consultas históricas
Registrar datos del productor
Registrar el pago de la papa
Registrar el programa de entregas
Registrar inspección al lugar de cultivo
Registrar la cancelación de la operación de canje
Registrar la compra de sobrante de papa
Registrar la entrega de papa
Registrar la entrega de semillas en concepto de canje
Registrar la inspección de papa
Registrar la orden de carga
Registrar las horas de riego
Registrar las precipitaciones ocurridas
Registrar funguicidas, insecticidas y/o herbicidas
Registrar los turnos para retirar las semillas
Registrar salidas de dinero
Registrar siembra de papa

A partir de los EF, aplicando la heurística de generación, se ha obtenido una lista de 112 requisitos.

El objetivo general del sistema es la administración de todas las operaciones que ocurren en el marco de la relación del acopiador con los productores de papa. Aplicando la primera etapa de la estrategia de asignación de prioridades, se divide el objetivo general en sub-objetivos. En este caso, fue suficiente un único nivel de descomposición para lograr la asignación de prioridades consistentes. Esta división puede observarse en la Figura 3.

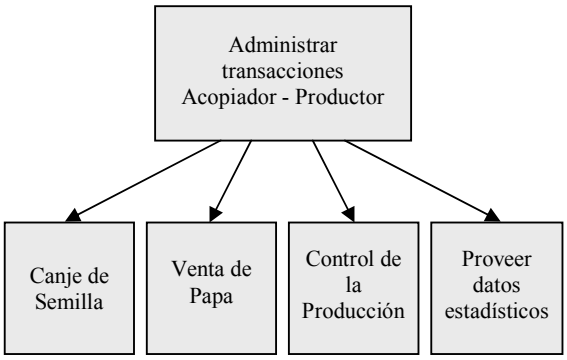

Figura 3. Jerarquía de objetivos del sistema

Una vez finalizada la división, como lo indica el segundo paso de la estrategia, se procede a la asignación de prioridades a cada sub-objetivo, habiéndose establecido una escala de 1 a 4, correspondiendo el menor valor a la mayor prioridad, y se obtuvo el siguiente resultado (paso ***iii***):

Prioridad	Sub-Objetivo
1	Control de la Producción
2	Venta de Papa
3	Canje de Semilla
4	Proveer Datos Estadísticos

En la Tabla 2, se presenta la asociación de cada uno de estos sub-objetivos con los EF, y se indica en la última columna la cantidad de requisitos asociados a cada escenario futuro.

La prioridad de cada uno de los sub-objetivos se traslada a los requisitos a los que dio lugar cada uno de los EF, como indica el paso ***v*** de la estrategia.

Como puede observarse en la Tabla 2, existe un escenario futuro común a dos sub-objetivos: **Iniciar la operación de venta de papa**. Como ambos sub-objetivos tienen prioridades diferentes, se debe determinar qué requisitos corresponden a cada sub-objetivo, para asignarles la prioridad correspondiente (paso ***vi***).

Los requisitos en negrita en la Figura 4 corresponden a la prioridad 1, es decir al sub-objetivo Control de Producción, mientras el resto corresponde a la prioridad 2, del sub-objetivo Venta de Papa.

Al considerar las prioridades de los sub-objetivos de prioridad 1 y 2, surge la necesidad de construir el escenario futuro de transición que se presenta en la Figura 5, según el paso ***ix***.

Tabla 2. Relación entre sub-objetivos del sistema y Escenarios Futuros

	Sub-Obj.	Escenarios Futuros	Cantidad Req.
Prioridad 1	Control de la Producción	Registrar datos del productor	4
		Iniciar la operación de venta de papa	8
		Registrar inspección al lugar de cultivo	4
		Registrar siembra de papa	2
		Registrar los funguicidas, insecticidas y/o herbicidas utilizados	3
		Registrar las precipitaciones ocurridas	2
		Registrar las horas de riego	3
		Chequear las variables de la producción	3
Prioridad 2	Venta de Papa	**Iniciar la operación de venta de papa**	9
		Registrar el programa de entregas	2
		Registrar la entrega de papa	19
		Registrar la inspección de papa	3
		Registrar la compra de sobrante de papa	4
		Registrar el pago de la papa	14
Prioridad 3	Canje de Semilla	Iniciar la operación de canje de semilla de papa	7
		Registrar la cancelación de la operación de canje	8
		Registrar los turnos para retirar las semillas	1
		Registrar la orden de carga	3
		Registrar la entrega de semillas de papa en concepto de canje	3
Prioridad 4	Proveer Datos Estadísticos	Realizar consultas actuales	5
		Realizar consultas históricas	5
		Pedir estadísticas	8

Este escenario de transición contiene al actor sistema llevando a cabo las actividades correspondientes al sub-objetivo de mayor prioridad, mientras que el resto de las actividades son efectuadas en forma manual, retrotrayéndose a la situación modelada por los escenarios actuales. Como consecuencia, el escenario futuro de transición resultante representa el momento en el que se ha finalizado la puesta en servicio del software del primer sub-objetivo y no se ha puesto aún en servicio el software para el segundo sub-objetivo.

Escenario Futuro

Título: INICIAR LA OPERACIÓN DE VENTA DE PAPA
Objetivo: Acordar la producción de papa y sus condiciones.
Contexto:
Ubicación Geográfica: Oficina administrativa.
Ubicación Temporal:
Precondición: El Productor debe conocer globalmente las recomendaciones relativas a la producción
Recursos: listado de recomendaciones relativas a la producción, lugares de cultivo de El Productor
Actores: El Operador de Venta, El Productor, El Acopiador, Sistema.
Episodios:

1. **SI** El Productor no está registrado en el sistema, **ENTONCES REGISTRAR DATOS DEL PRODUCTOR**.
2. El Productor informa el o los lugares de cultivo donde se producirá la papa.
3. El Operador de Venta registra en el sistema los lugares de cultivo indicados.
4. El Productor especifica la cantidad de papa que entregará.
5. El Operador de Venta registra en el sistema la cantidad especificada.
6. **FIJAR EL PRECIO DE LA PAPA**
7. El Operador de Venta registra en el sistema el precio por tonelada.
8. El Acopiador y El Productor pactan las recomendaciones relativas a la producción que se aplican al caso.
9. El Operador de Venta registra en el sistema las recomendaciones relativas a la producción.
10. El sistema habilita a dar de alta las inspecciones en el lugar de cultivo.
11. El Acopiador y El Productor pactan las compensaciones adicionales que se aplicarán, los descuentos y los períodos bonificados.
12. El Operador de Venta registra en el sistema las compensaciones, descuentos y períodos bonificados pactados.
13. El Operador de Venta solicita al sistema que imprima el Contrato de adquisición y producción de papa
14. El Acopiador y El Productor firman el Contrato de adquisición y producción de papa

Excepciones:

- **El sistema debe registrar los lugares de cultivo donde se producirá la papa**
- El sistema debe registrar el precio por tonelada al cual El Acopiador pagará la papa
- **El sistema debe registrar las recomendaciones relativas a la producción**
- **El sistema debe habilitar el alta de las inspecciones en el lugar de cultivo donde se producirá la papa**
- El sistema debe registrar las compensaciones adicionales que se aplicarán
- El sistema debe registrar los descuentos que se realizarán
- El sistema debe registrar cuáles serán los períodos bonificados
- El sistema debe imprimir el Contrato de adquisición y producción de papa

Figura 4. Escenario futuro INICIAR LA OPERACIÓN DE VENTA DE PAPA

Escenario Futuro de Transición

Título: INICIAR EL CONTROL DE LA PRODUCCIÓN
Objetivo: Acordar la producción de papa y sus condiciones.
Contexto:
Ubicación Geográfica: Oficina administrativa.
Ubicación Temporal:
Precondición: El Productor debe conocer globalmente las recomendaciones relativas a la producción
Recursos: listado de recomendaciones relativas a la producción, lugares de cultivo de El Productor, Formulario preimpreso del Contrato de adquisición y producción de papa
Actores: El Operador de Venta, El Productor, El Acopiador, Sistema.
Episodios:

Nuevo recurso

1. **SI** El Productor no está registrado en el sistema, **ENTONCES REGISTRAR DATOS DEL PRODUCTOR**.
2. El Productor informa el o los lugares de cultivo donde se producirá la papa.
3. El Operador de Venta registra en el sistema los lugares de cultivo indicados.
4. El Productor especifica la cantidad de papa que entregará.
5. Se elimina el episodio 5
6. **FIJAR EL PRECIO DE LA PAPA**
7. Se elimina el episodio 7
8. El Acopiador y El Productor pactan las recomendaciones relativas a la producción que se aplican al caso.
9. El Operador de Venta registra en el sistema las recomendaciones relativas a la producción.
10. El sistema habilita a dar de alta las inspecciones en el lugar de cultivo.
11. El Acopiador y El Productor pactan las compensaciones adicionales que se aplicarán, los descuentos y los periodos bonificados.
12. Se elimina el episodio 12
13. El Operador de Venta completa en forma manual el formulario preimpreso del contrato de adquisición y producción de papas.
14. El Acopiador y El Productor firman el Contrato de adquisición y producción de papa

Se modifica el episodio 13, eliminando el actor sistema.

Excepciones:

- El sistema debe registrar los lugares de cultivo donde se producirá la papa
- El sistema debe registrar las recomendaciones relativas a la producción
- El sistema debe habilitar el alta de las inspecciones en el lugar de cultivo donde se producirá la papa

Figura 5. Escenario futuro de transición

Se debe aclarar que en este caso de estudio todos los RNF detectados fueron convertidos en servicios del sistema durante la construcción de los EF, no permaneciendo ninguno con naturaleza no funcional al arribar a la etapa de explicitar requisitos.

7. Conclusiones

La estrategia para extraer requisitos de software de los escenarios ayuda a redactar un SRS no ambiguo cubriendo un aspecto no tratado en detalle en la literatura: cómo pasan los requisitos desde los modelos construidos al SRS.

Se ha presentado un mecanismo para la compatibilización de prioridades de requisitos, basado en la descomposición de objetivos en sub-objetivos y utilizando a los escenarios como puente para trasladar prioridades de los sub-objetivos a los requisitos. Este mecanismo facilita la asignación de prioridades individualmente a los requisitos, proveyendo de un contexto para tal asignación.

La estrategia propuesta ha podido aplicarse a diferentes casos de estudio, obteniendo en todos los casos un conjunto de sub-objetivos con diferentes prioridades, a partir de los cuales es posible asignar prioridades a los requisitos originales.

Se pudo observar que en la mayoría de los casos estudiados la descomposición de los objetivos conduce a la construcción de escenarios de transición a partir de escenarios comunes a más de un sub-objetivo. Estos escenarios son una suerte de escenarios de transición entre dos áreas del proceso del negocio. Sin embargo, esta situación ocurre en pocos escenarios por caso analizado.

A diferencia de la mayoría de las técnicas de asignación de prioridades a requisitos mencionadas en la sub-sección 2.1, el mecanismo propuesto facilita dicha actividad al promover que los involucrados asignen prioridades a los sub-objetivos del sistema de software, para que luego las mismas sean trasladadas a los requisitos. Es de observar que existe una diferencia notable de magnitud entre los sub-objetivos y los requisitos.

Algunos atributos incluidos en el meta-modelo de Especificación de Requisito responden a la posibilidad de calcular la prioridad teniendo en cuenta otros aspectos ya mencionados en la literatura (ver sección 2.1). La incorporación de ellos al mecanismo de asignación de prioridades propuesto será un aspecto a considerar en futuros trabajos.

Asimismo, se proyecta comparar el mecanismo presentado en este artículo con las técnicas existentes con el fin de evaluar cuantitativamente las diferencias de esfuerzo que demandan.

Referencias

[1] J.H. Doorn, G.D.S. Hadad, and G.N. Kaplan, "Comprendiendo el Universo de Discurso Futuro", *WER'02 - Workshop on Requirements Engineering*, Valencia, España, 2002, pp. 117-131.

[2] J.C.S.P. Leite, J.H. Doorn, G.N. Kaplan, G.D.S. Hadad, and M.N. Ridao, "Defining System Context using Scenarios", *Perspectives on Software Requirements*, Kluwer Academic Publishers, cap.8, 2004, pp. 169-199.

[3] V. Ahl, *An Experimental Comparison of Five Prioritization Methods. Master's Thesis*, School of Engineering, Blekinge Institute of Technology, Suecia 2005.

[4] B.W. Boehm, and H. In, "Identifying Quality-Requirement Conflicts", *IEEE Software*, Vol.13, Nº2, 1996, pp. 25-35.

[5] L. Chung, B.A. Nixon, and E. Yu, *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers, 2000.

[6] A. Davis, "The Art of Requirements Triage", *IEEE Computer*, Vol.36, Nº3, 2003, pp. 42-49.

[7] S. Easterbrook, "Handling conflict between domain descriptions with computer-supported negotiation", *Knowledge Acquisition: An International Journal*, Vol.3, 1991, pp. 255-289.

[8] R.E. Fairley, and R.H. Thayer, "The Concept of Operations: The Bridge from Operational Requirements to technical Specifications", *Software Requirements Engineering*, Richard H. Thayer & Merlin Dorfman eds., IEEE Computer Society Press, 2º edición, Los Alamitos, CA, 1997, pp. 73-83.

[9] G.D.S. Hadad, *Uso de Escenarios en la Derivación de Software. Tesis doctoral*, Facultad de Ciencias Exactas de la Universidad Nacional de La Plata, Argentina, 2008.

[10] *IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications (ANSI)*, IEEE, Nueva York, 1998.

[11] *IEEE Std P1233/D3-1995, IEEE Guide for Developing System Requirements for Specifications*, IEEE, Nueva York, 1995.

[12] *ISO 9126:2001, International Standard ISO/IEC 9126*. Information technology - Software product evaluation - Quality characteristics and guidelines for their use, International Organization for Standardization, International Electrotechnical Commission, 2001.

[13] F. Moisiadis, "Prioritising Scenario Evolution". *International Conference on Requirements Engineering (ICRE 2000)*, 2000.

[14] I. Sommerville, and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, John Wiley & Sons, cap. 1-2 , 1997.

[15] R.R. Young, *The Requirements Engineering Handbook*, Artech House, Norwood, MA, 2004.

[16] J.W. Brackett, *Software Requirements*, (SEI-CM-19-1.2, ADA235642), Pittsburgh, Software Engineering Institute, Carnegie Mellon University, 1990.

[17] T.L. Saaty, *The Analytic Hierarchy Process*. McGraw-Hill, 1980.

[18] J. Karlsson, C. Wohlin, and B. Regnell, "An evaluation of methods for prioritizing software requirements", *Information and Software Technology*, Vol.39, Nº14-15, 1998, pp. 939-947.

[19] R. Zultner, "Quality Function Deployment (QFD) for Software: Structured Requirements Exploration", *Total Quality Management for Software*, editores Schulmeyer & McManus, NY: Van Nostrand Reinhold, 1992, pp. 297-317.

[20] D. Leffingwell, and D. Widrig, *Managing Software Requirements - A unified approach*. Addison-Wesley Object Technology Series, 2º edición, 2003.

[21] B. Boehm, and R. Ross, *Theory-W Software Project Management: Principles and Examples*, IEEE TSE, Vol.15, Nº4, 1989, pp. 902-916.

[22] J. Park, D. Port, and B. Boehm, "Supporting Distributed Collaborative Prioritization for Win-Win Requirements Capture and Negotiation", *International Third World Multi-conference on Systemics, Cybernetics and Informatics (SCI'99)*, Orlando, FL, Vol.2, 1999, pp. 578-584.

[23] K. Beck, and C. Andres, *Extreme Programming Explained: Embrace Change*, Boston, MA: Addison-Wesley, 2º edición, 2004.

[24] D. Greer, D. Bustard, and T. Sunazuka, "Prioritisation of system changes using cost-benefit and risk assessments", *Fourth IEEE International Symposium on Requirements Engineering*, 1999, pp. 180-187.

[25] K.E. Wiegers, *Software Requirements*, Redmond, Microsoft Press, 2º edición, 1999.

[26] D. Greer, "Requirements Prioritisation for Incremental and Iterative Development", *Requirements Engineering for Sociotechnical Systems,* Information Science Publishing, editores Maté & Silva, cap.VII, 2005, pp. 100-118.

[27] G. Ruhe, A. Eberlein, and D. Pfahl, "Quantitative WinWin: a new method for decision support in requirements negotiation", *14th International Conference on Software Engineering and Knowledge Engineering (SEKE'2002)*, ACM, Nueva York, 2002, pp. 159-166.

[28] F. Moisiadis, "A Requirements Prioritisation Tool", *6th Australian Workshop on Requirements Engineering (AWRE 2001)*, Sydney, Australia, 2001.

[29] J. Mylopoulos, L. Chung, and B.A. Nixon, "Representing and Using Nonfunctional Requirements: A Process-Oriented Approach", *IEEE TSE*, Vol.18, Nº6, 1992, pp. 483-497.

[30] G. Kotonya, and I. Sommerville, *Requirements Engineering: Processes and Techniques*, John Wiley & Sons, 1998.

[31] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition", *Science of Computer Programming*, Vol.20, 1993, pp. 3-50.

[32] C. Potts, "Using Schematic Scenarios to Understand User Needs", *DIS'95 - Symposium on Designing Interactive Systems: Processes, Practices and Techniques*, ACM Press, University of Michigan, 1995, pp. 247-256.

[33] A. Cockburn, "Using Goal-Based Use Cases", *Journal of Object-Oriented Programming (JOOP)*, Vol.10, Nº 7, 1997, pp. 56-62.

[34] A. Davis, and D. Leffingwell, "Making Requirements Management Work For You", *Crosstalk, The Journal of Defense Software Engineering*, Vol.12, Nº4, 1999.

[35] P. Sawyer, and G. Kotonya, "Software Requirements", *SWEBOK, Guide to the Software Engineering Body of Knowledge*, editores P. Bourque y R. Dupuis, IEEE Computer Society, Los Alamitos, CA, cap.2, 2004, pp. 2-1 – 2-17.

[36] J. Whitten, L. Bentley, and K. Dittman, *Systems Analysis and Design Methods*, Mc Graw-Hill / Irwin, 6º edición, cap. 6, 2003.

[37] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering. Springer Science*, 2º edición, 2005.

[38] J.C.S.P. Leite, J.H. Doorn, G.D.S. Hadad, and G.N. Kaplan, "Scenario Inspections", *Requirements Engineering Journal*, Vol.10, Nº1, Springer-Verlag, 2005, pp. 1-21.

[39] A. van Lamsweerde, "Inferring Declarative Requirements Specifications from Operational Scenarios", *IEEE TSE*, Vol.24, Nº12, 1998, pp. 1089-1114.

[40] G.D.S. Hadad, J.H. Doorn, and G.N. Kaplan, "Explicitar Requisitos del Software usando Escenarios", *WER'09*, aceptado para publicación, Valparaíso, Chile, Julio 2009.