

Interacción de Atributos al Priorizar Requisitos a través de Objetivos

Graciela D. Hadad^{1,2}, Jorge H. Doorn¹, Claudia S. Litvak^{1,2}, Carlos Imparato¹

¹Escuela de Informática, Universidad Nacional del Oeste
Córdoba 1055. Merlo. Buenos Aires. (0220) 482-0799

²Facultad de Ingeniería y Tecnología Informática, Universidad de Belgrano
e-mail: gracielahadad@gmail.com, jdoorn@exa.unicen.edu.ar, claudialitvak@gmail.com,
imparatocarlos@gmail.com

Resumen

Una actividad que ejerce una fuerte influencia en la gestión de proyectos de software es establecer el orden de implementar los requisitos del software. Este ordenamiento determina la necesidad de recursos humanos y tecnológicos a lo largo del tiempo, el modo de puesta en operación parcial del software y su mantenimiento, entre otras actividades. Es uno de los motores de la evolución de los requisitos implementados y a implementar debido a realimentaciones de los usuarios al comenzar a utilizar los módulos instalados. Definir este orden es uno de los propósitos que detentan muchas técnicas de asignación de prioridades. Se ha observado que varias de estas técnicas no pueden ser aplicadas cuando se tiene un gran número de requisitos. Atendiendo a este problema, en proyectos anteriores se elaboró una propuesta para asignar prioridades utilizando los objetivos del software, de manera tal de transferir prioridades desde objetivos específicos hacia los requisitos, ya que el número de objetivos conforma un conjunto mucho menor que el de requisitos. La propuesta actual consiste en definir qué atributos deben utilizarse para establecer el orden de implementación de

los requisitos.

Palabras clave: Ingeniería de Requisitos, Priorizar Requisitos, Objetivos

Contexto

Esta propuesta es parte del proyecto de investigación “Tratamiento de los Factores Situacionales y la Completitud en la Ingeniería de Requisitos” de la UNO y del proyecto “Estrategia para mejorar la completitud de modelos orientados al cliente en la Ingeniería de Requisitos” de la UB.

Introducción

La mayoría de las estrategias de definición de requisitos incluyen la actividad de asignar prioridades a los requisitos. Esto es principalmente cierto cuando el proceso de desarrollo de software es iterativo e incremental y, por ende, es necesario establecer qué requisitos poner en servicio en cada iteración. La priorización puede desarrollarse de diversas maneras y con diversos propósitos [1]. La literatura ofrece una amplia gama de técnicas de priorización, desde muy sencillas que utilizan un solo atributo evaluado por un único actor, hasta técnicas más complejas

que involucran varios atributos y varios actores con distintos puntos de vista [2]. Muchas técnicas se aplican sobre requisitos, otras priorizan [3,4] casos de uso, historias de usuario, aspectos no funcionales, etc. Algunas técnicas manejan la interdependencia de requisitos y otras no. Algunas técnicas aplican escalas de valoración cuantitativas o cualitativas para valorizar atributos, otras combinan el uso de varias escalas [5]. Un aspecto a destacar es que las técnicas tienen propósitos diversos, ya que no siempre se han diseñado para establecer el orden de implementar requisitos. Algunas técnicas tienen su propia finalidad, básicamente: determinar la importancia relativa del requisito en un orden, o seleccionar el conjunto óptimo de requisitos a implementar en la siguiente iteración. En resumen, existen diferencias notorias entre las técnicas de priorización. A continuación se describen resumidamente algunas de ellas.

Dentro de las más simples está Ranking [2], que asigna un número único ordinal a cada requisito según su importancia para un cliente. En Asignación Numérica [6] un cliente agrupa a los requisitos en obligatorios, deseables y no esenciales. En MoSCoW [7] los clientes negocian para agrupar cualitativamente los requisitos en: fundamentales, importantes, deseables y a futuro. El Método de Wiegers [3] calcula una prioridad para cada requisito (caso de uso, característica del sistema u otro elemento) en base a la ponderación de cuatro atributos: beneficio para el cliente, penalidad por no incluir el requisito, costo de implementación y riesgo técnico. Los dos primeros son asignados por los clientes y los dos últimos por los desarrolladores. AHP (Analytic Hierarchy Process) [8] se basa en que un cliente compare los requisitos de a pares, estimando un valor relativo de

importancia entre ellos, generando una lista de requisitos con un valor proporcional de prioridad. Esta técnica es utilizada como parte de otras técnicas de priorización, como Costo-Valor y Quantitative WinWin. Sin embargo, varios estudios empíricos [9-11] han mostrado que su uso no es apropiado si se tiene un gran número de requisitos, por lo que se han propuesto variantes para disminuir el número de comparaciones [9]. La técnica 100 Point-Test [12] consiste en darle 100 puntos a cada involucrado para que los distribuya entre los requisitos más importantes para él. Se obtiene una lista de requisitos ordenada por los puntos asignados entre todos ellos. En Top-Ten Requirements [13] cada cliente elige los diez requisitos que considera de mayor prioridad y luego, se arma una lista con los requisitos más importantes (sin ordenar), balanceando la cantidad de requisitos elegidos por cada cliente. Costo-Valor [14] estima para cada requisito, usando AHP, el beneficio por parte del cliente y el costo por parte del desarrollador y se seleccionan los requisitos con menor índice de costo y mayor índice de beneficio. VOP (Value-Oriented Prioritization) [15] utiliza una fórmula ponderada para calcular la prioridad de cada requisito en función del valor y del riesgo para el negocio, calificados por consenso entre los clientes, y se obtiene un ranking de requisitos con el orden de implementación. Requirements Triage [16] establece jerarquías de requisitos y sus dependencias; los desarrolladores estiman el esfuerzo de implementar cada requisito y también el esfuerzo por implementar un conjunto de requisitos dependientes; los clientes establecen por votación la importancia relativa de cada requisito. En base al esfuerzo e importancia estimados, los clientes, desarrolladores y representantes

financieros deciden qué requisitos implementar. En Easy WinWin [17] participan clientes, usuarios y desarrolladores agrupando los requisitos en cuatro grupos según su importancia para el negocio y su dificultad de implementación, mediante el modelo de negociación WinWin. Quantitative WinWin [18], variante de la anterior, combina técnicas cualitativas con cuantitativas, reflejando distintos puntos de vista y preferencias, necesidad de recursos e interdependencia de requisitos, y usa AHP al estimar la importancia relativa de clases de requisitos. Se determina si el esfuerzo de implementar el conjunto de requisitos es aceptable. Si no lo es, se modifica el conjunto y se iteran los pasos. SQFD (Software Quality Function Deployment) [19] propone definir los requisitos en los cuales invertir los recursos durante el desarrollo. Para ello, considera tipos de clientes y su importancia para la toma de decisión, las necesidades del negocio y sus interdependencias, y el grado de contribución de los requisitos a las necesidades. En el Juego de Planeamiento [4] el cliente agrupa historias de usuario según el valor que aportan al negocio y se les asigna un orden en el grupo. Pirogov [20] automatiza gran parte del proceso de priorización, maneja dependencias entre requisitos, permite la negociación entre los involucrados, definiendo varios criterios para priorizar.

En proyectos anteriores [1], se determinó que existen tres propósitos básicos para priorizar requisitos: i) establecer la preferencia o necesidad relativa entre requisitos (ej. AHP, 100 Point Test), ii) seleccionar requisitos candidatos a ser incluidos en el producto software (ej: Easy Win-Win, Costo-Valor), y iii) definir un orden secuencial en los requisitos (ej: VOP, Pirogov, Método de Wieggers). Principalmente, este

último propósito atiende al orden de implementación de los mismos y, para ello, se debería tener la visión de los clientes y desarrolladores, y contemplar la interdependencia de requisitos. Algunas técnicas atienden más de un propósito, como Quantitative Win-Win, Requirements Triage y Juego de Planeamiento; este último selecciona qué requisitos considerar y cuáles no, y luego determinar el orden de implementación de aquellos seleccionados. Asimismo, se detectó que en algunos casos el propósito de algunas de ellas no es coherente con su uso (Tabla 1) debido al manejo poco apropiado de algunas características. Por ejemplo, VOP no contempla el punto de vista del desarrollador ni la dependencia entre requisitos para establecer apropiadamente el orden de implementación de requisitos.

Tabla 1. Técnicas de priorización con uso no coherente respecto al propósito



De la revisión realizada en la literatura [1], se confirmó que aquellas técnicas que incluían múltiples atributos y múltiples puntos de vista mostraban mejores resultados para establecer el orden de implementación de requisitos.

Líneas de Investigación, Desarrollo e Innovación

En base al estudio de estas técnicas de priorización y los problemas observados, se propone mejorar una técnica ya diseñada [21] que se aplica en el contexto de una estrategia de Ingeniería de

Requisitos basada en modelos escritos en lenguaje natural [22], de manera tal que la misma contemple la calificación de varios atributos para calcular la prioridad, sin obstaculizar su facilidad de aplicación.

La técnica parte de un paso previo de creación de un árbol de objetivos del software, y consiste en la asignación de prioridades a los sub-objetivos por parte de los clientes, la transferencia de prioridades de los sub-objetivos a los escenarios que los satisfacen y, finalmente, la transferencia de prioridades desde los escenarios a los requisitos extraídos de cada escenario. Se aplica tanto a requisitos funcionales como no funcionales. En el caso que un escenario satisfaga más de un sub-objetivo con distinta prioridad, se establece para cada requisito extraído del escenario qué sub-objetivo satisface. La facilidad de la técnica reside en que el cliente solo asigna prioridades a los sub-objetivos, que son una cantidad bastante menor que el de requisitos, y la transferencia de dichas prioridades hacia los requisitos las realiza el ingeniero de requisitos.

Cuando se asignan prioridades a los objetivos del sistema, dicha prioridad está asociada al concepto de orden de implementación. La técnica usa como único atributo a calificar la prioridad y solo considera el punto de vista de un cliente. La interdependencia de requisitos está contemplada dado que la transferencia de prioridad se hace al escenario, el cual relaciona un conjunto de requisitos para satisfacerlo. Para que el uso de la técnica sea coherente con su propósito: *definir el orden de implementación de los requisitos*, se debería descomponer la prioridad en atributos que reflejen características que ayuden en la decisión de cuál requisito implementar antes que otro, e incluir el punto de vista de los desarrolladores, quienes pueden ayudar a determinar cómo

reducir costos y tiempos al momento de implementar un conjunto de requisitos, contemplando las limitaciones impuestas al proyecto.

En principio, se usaron dos atributos: urgencia para el negocio (necesidad del cliente) y la criticidad (importancia relativa en cuanto a la satisfacción de los objetivos del sistema). Para facilitar la calificación de cada requisito respecto a estos atributos, los clientes califican los sub-objetivos según la urgencia, la que se traslada a los requisitos, estos se agrupan por sub-objetivo que satisfacen, y los clientes valoran dentro de cada grupo la criticidad del requisito respecto a dicho sub-objetivo. La técnica refinada fue probada en un caso real, donde se comprobó que el cliente no tenía dificultades en tratar grupos acotados de requisitos para valorizar su criticidad.

Como efecto colateral de la prueba realizada, se detectó que algunos pocos requisitos no satisfacían ningún objetivo, concluyéndose que estos habían sido omitidos. Dado lo cual, se incorporó a la técnica un refinamiento iterativo del árbol de objetivos.

Se ha iniciado la evaluación de la relación entre los atributos volatilidad y riesgo para el negocio, desde la perspectiva de los objetivos del sistema y de los propios requisitos, con el fin de incorporarlos en la priorización. Al aplicarlos en un caso, se observó que el riesgo de negocio asociado a los requisitos era en todos los casos menor o igual que el riesgo asociado al objetivo que satisface, mientras que la volatilidad asociada a los requisitos era siempre mayor o igual que la asociada al objetivo. Esta relación solo se cumplía para los requisitos funcionales. Se debe evaluar si esta relación puede facilitar la priorización al permitir calificar ciertos atributos a través de los objetivos, en vez de calificar individualmente requisitos.

Resultados y Objetivos

La técnica de priorización diseñada permite asignar prioridades cuando se tiene una gran cantidad de requisitos que deben implementarse en sucesivos incrementos. Este hecho ha sido justamente cuestionado por estudios empíricos en la aplicación de varias técnicas propuestas en la literatura [5,11], estableciéndose la inviabilidad de algunas de ellas en su puesta en práctica en proyectos de cierta envergadura. Este es el motor conductor en las mejoras que se están incorporando a la técnica, teniendo en consideración que el facilitar la tarea de los clientes al calificar requisitos logra mejores respuestas por parte de ellos.

Se realizarán nuevas pruebas sobre el refinamiento actual de la técnica y se estudiará en profundidad la relación entre los atributos y los objetivos, y cómo esta relación puede influir en facilitar la priorización. Asimismo, se propone definir criterios para el tratamiento de los atributos en base al tipo de proyecto y al contexto de desarrollo.

Formación de Recursos Humanos

En el proyecto vinculado a la línea de investigación presentada aquí, se están formando dos becarios de grado de la UNO. En la misma línea, el año anterior se inició a la investigación un becario de grado de UNLaM, quien realizó publicaciones en congresos. Asimismo, esta línea involucra temas de las tesis de maestría en desarrollo en UNLaM: “Estrategia de Requisitos adaptable según factores de situación” de Ing. Viviana Ledesma y “Estrategia para la mitigación de la volatilidad de los requisitos en desarrollos multi-site” de Ing. Juan Pablo Mighetti. El proyecto reúne el trabajo de tesis doctoral: “Gestión de de la completitud en la Ingeniería de

Requisitos” de la Mg. Claudia Litvak en UNLP.

Referencias

- [1] Hadad G, Riera G, Doorn J. “Priorizar Requisitos: un Estudio sobre sus Propósitos”, 2º CONAIISI, 2014, pp.953-962
- [2] Berander P, Andrews A. “Requirements Prioritization”, Engineering and Managing Software Requirements, A.Aurum, C.Wohlin (eds.), Springer, ch.4, 2005, pp.69-94
- [3] Wieggers KE. “First Thing First: Prioritizing Requirements”, Software Development, 1999, 7(9):24-30
- [4] Beck K, Andres C. Extreme Programming Explained: Embrace Change, Addison-Wesley, 2ºed., 2004
- [5] Ma Q. The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review, Master Thesis, Auckland University of Technology, 2009
- [6] Brackett JW. “Software Requirements”, SEI-CM-19-1.2, ADA235642. Carnegie Mellon University, 1990
- [7] Clegg D, Barker R. Case Method Fast-Track, A RAD Approach, Addison-Wesley, 1994
- [8] Saaty TL. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, McGraw-Hill, 1980
- [9] Karlsson J, Wohlin C, Regnell B. “An evaluation of methods for prioritizing software requirements”, Information and Software Technology, 1998, 39(14-15):939-947
- [10] Maiden N, Ncube C. “Acquiring COTS software selection requirements”, IEEE Software, 1998, 15(2):46-56
- [11] Lehtola L, Kauppinen M. “Empirical evaluation of two requirements prioritization methods in product development projects”, European Software Process Improvement Conference, Springer-Verlag, 2004, pp.161-170
- [12] Leffingwell D, Widrig D. Managing Software Requirements - A unified approach, Addison-Wesley, 2º ed, 2003
- [13] Lausen S. Software requirements – styles and techniques, Pearson Education, Essex, 2002
- [14] Karlsson J, Ryan K. “A Cost-Value Approach for Prioritizing Requirements”, IEEE Software, 1997, 14(5):67-74
- [15] Azar J, Smith RK, Cordes D. “Value-Oriented Requirements Prioritization in a Small Development Organization”, IEEE Software, 2007, pp.32-37
- [16] Davis A. “The Art of Requirements Triage”, IEEE Computer, 2003, 36(3):42-49
- [17] Grünbacher P. “Collaborative Requirements Negotiation with Easy WinWin”, 2nd Intl Workshop on the Requirements Engineering Process, IEEE Computer Society, 2000, pp.954-960
- [18] Ruhe G, Eberlein A, Pfahl D. “Quantitative WinWin: a new method for decision support in requirements negotiation”, 14th Intl Conference on Software Engineering and Knowledge Engineering, ACM, 2002, pp.159-166
- [19] Zultner R. “Quality Function Deployment for Software: Structured Requirements Exploration”, Total Quality Management for Software, Schulmeyer, McManus (eds) 1992, pp.297-317
- [20] Duan C, Laurent P, Cleland-Huang J, et al. “Towards automated requirements prioritization and triage”, REJ, 2009, 14(2):73-89
- [21] Hadad G, Doorn J, Ridao M, Kaplan G. “Facilitando la asignación de Prioridades a los Requisitos”, 12th WER, Valparaíso, 2009, pp.75-84
- [22] Leite J, Doorn J, Kaplan G, Hadad G, Ridao M. “Defining System Context using Scenarios”, Perspectives on Software Requirements, Kluwer Academic Publishers, J.Leite, J.Doorn (eds), cap.8, 2004, pp.169-199