

Análisis de Impacto de Cambios en Requisitos Software

Posgrado en Ciencias y Tecnologías de la Información

Marzo del 2012.

1. Responsables

Dra. Angelina Espinoza Limón

Escuela Universitaria de Informática
System and Software Technologies Group (SYST)
Universidad Politécnica de Madrid (UPM)
Email: a.espinoza@upm.es

M. en C. Alfonso Martínez Martínez

Laboratorio de Investigación en Informática Médica (LIIM)
Área de Procesamiento Digital de Señales e Imágenes Biomédicas
Departamento de Ing. Eléctrica
UAM-Iztapalapa
Email: almm@xanum.uam.mx

2. Perfil deseable del alumno

Tener conocimientos de Ingeniería de Software relacionados a:

- Procesos y ciclo de desarrollo de software
- Trazabilidad de Requisitos
- Modelado con UML
- Eclipse Modeling Framework y Graphical Modeling Framework

3. Contexto y problemática

El análisis de impacto de cambios es fundamental en la evolución de sistemas software, ya que permite determinar en términos cuantitativos el impacto de un cambio en un requisito de software, hacia los subsecuentes modelos del sistema (clases, arquitectura, de implementación, etc).

Trazabilidad de requisitos de software es un mecanismo fundamental en el desarrollo de proyectos software de alta calidad. Trazabilidad es un mecanismo para describir y seguir la vida de un artefacto de software (requisitos, clases, componentes, tests, etc) a través del proceso de desarrollo completo, así mismo también es un mecanismo para modelar explícitamente las relaciones entre todos los artefactos de desarrollo del sistema software (consultar [3, 4, 7, 8, 12]).

Trazabilidad actualmente es considerada una tarea de apoyo en el desarrollo de aplicaciones software requerida por el estándar ISO 12207, e incluida como parte de un proceso maduro por CMMi. Este

mecanismo permitirá apoyar a un número de procesos de desarrollo de software, como los son gestión del impacto de cambios (revisar [1, 5, 11]). Así mismo es un mecanismo usado en Ingeniería Dirigida por Modelos [2, 6] e Ingeniería de Líneas de Productos Software [9] y para apoyar en la evolución de los sistemas software [10]. La razón es que proporciona una visibilidad en términos cuantitativos de:

- Avance real de la realización de los requisitos software
- Implementación de los requisitos software en los artefactos desarrollados en subsecuentes etapas de desarrollo (diseño, implementación, testing)
- Grado de dependencia entre los artefactos de desarrollo (fundamental en análisis de impacto)

Sin embargo, es necesario un apoyo automatizado para realizar trazabilidad, de otra forma se convierte en una tarea exhaustiva además de costosa.

Por lo tanto, la realización de este trabajo se centrará en proporcionar: 1) un modelo de trazabilidad para apoyar el análisis de impacto de cambios en requisitos software, 2) Un algoritmo basado en el modelo, para determinar el impacto. La viabilidad de la propuesta se mostraría mediante un prototipo de herramienta, que facilite tanto el proceso de creación de links de trazabilidad, como el análisis del impacto de un cambio.

4. Objetivos

4.1. Objetivo General

Este trabajo, tiene como objetivo proporcionar un modelo de apoyo en la realización del análisis de impacto de cambios en requisitos software. El modelo de apoyo al análisis de impacto se basará fuertemente en trazabilidad.

4.2. Objetivos específicos

Para lograr el objetivo general se proponen los siguientes objetivos específicos:

1. Definir un modelo de trazabilidad para apoyar el análisis de impacto de cambios en requisitos software.
2. Proporcionar un algoritmo basado en el modelo, para determinar el impacto de un cambio en un requisito basado en trazabilidad.
3. Desarrollar un prototipo de herramienta que facilite tanto el proceso de creación de links de trazabilidad, como el análisis del impacto de un cambio.
4. Probar la viabilidad de la propuesta en un entorno industrial, mediante casos de estudio.

5. Metodología

1. Análisis breve de literatura relacionada a trazabilidad de requisitos de software.
2. Análisis detallado de literatura y experiencias relacionadas a análisis de impacto centrado en trazabilidad.
3. Desarrollo del modelo de trazabilidad para apoyar el análisis de impacto de cambios en requisitos software, con base en:

- a)* el análisis previo realizado sobre el estado del arte y de la práctica sobre trazabilidad y análisis de impacto de cambios
 - b)* los requisitos que plantean los objetivos de esta tesis.
- 4. Proporcionar un algoritmo basado en el modelo para determinar el impacto de un cambio en un requisito, basado en trazabilidad.
- 5. Estudio de herramienta de modelado y desarrollo de herramienta prototipo
- 6. Validación del modelo en casos de estudio

6. Resultados esperados

- Obtener un modelo pragmático y simple que apoye en la estimación del impacto de un cambio en los requisitos software, en las etapas subsecuentes del ciclo de desarrollo de software.
- Proporcionar un algoritmo que basado en el modelo anterior sea capaz de determinar el impacto de un cambio en un requisito software.
- Producir un apoyo automatizado del modelo, con el objeto de disminuir el esfuerzo adicional que supone el uso del propio modelo.
- Obtener resultados de validación del modelo mediante la ejecución de casos de estudio.

7. Calendarización de actividades

Actividades	Trimestre 1				Trimestre 2				Trimestre 3			
Análisis breve de literatura relacionada a trazabilidad de requisitos de software.												
Análisis detallado de literatura y experiencias relacionadas a análisis de impacto centrado en trazabilidad.												
Desarrollo del modelo de trazabilidad para apoyar el análisis de impacto de cambios en requisitos software												
Estudio de las aplicaciones para desarrollar el prototipo de herramienta.												
Protocolo de investigación												
Desarrollo del modelo de trazabilidad para apoyar el análisis de impacto de cambios en requisitos software												
Desarrollo del algoritmo basado en el modelo para determinar el impacto de un cambio en un requisito, basado en trazabilidad.												
Primera versión de prototipo de herramienta												
Reporte en formato de artículo												
Segunda versión de prototipo de herramienta												
Validación mediante en un caso de estudio del modelo y ajustarlo con base en los resultados del caso de estudio.												
Primer borrador de tesis												

8. Infraestructura necesaria y disponible

1. Computadora con acceso a Internet
2. Acceso a las bases de datos digitales ACM, IEEE, Springer Verlag

9. Lugar de realización

Laboratorio de Investigación en Informática Médica, T-226.

Referencias

- [1] Jane Cleland-Huang, Carl K. Chang, and Jeffrey C. Wise. Automating performance-related impact analysis through event based traceability. *Requir. Eng.*, 8(3):171–182, 2003.
- [2] Nikolaos Drivalos, Dimitrios S. Kolovos, Richard F. Paige, and Kiran J. Fernandes. *Software Language Engineering*, chapter Engineering a DSL for Software Traceability, pages 151–167. LNCS, Springer-Verlag, Berlin, Heidelberg, 2009.

- [3] O.C.Z. Gotel and C.W. Finkelstein. An analysis of the requirements traceability problem. In *Proceedings of the International Conference on Requirements Engineering (RE)*, pages 94–102, Colorado Springs, Co., 1994. IEEE Computer Society Press.
- [4] Patricia Lago, Henry Muccini, and Hans van Vliet. A scoped approach to traceability management. *J. Syst. Softw.*, 82(1):168–182, 2009.
- [5] J.S. O’Neal and D.L. Carver. Analyzing the impact of changing requirements. In *Software Maintenance, 2001. Proceedings. IEEE International Conference on*, pages 190–195. IEEE, 2001.
- [6] Richard Paige, Nikolaos Drivalos, Dimitrios Kolovos, Kiran Fernandes, Christopher Power, Goran Olsen, and Steffen Zschaler. Rigorous identification and encoding of trace-links in model-driven engineering. *Software and Systems Modeling*, 10:469–487, 2011. 10.1007/s10270-010-0158-8.
- [7] B. Ramesh and M. Jarke. Toward reference models for requirements traceability. *IEEE Transactions on Software Engineering*, 27(1):58–93, Jan 2001.
- [8] Bala Ramesh, Lt Curtis Stubbs, Michael Edwards, and Timothy Powers. Lessons learned from implementing requirements traceability. *Crosstalk, Journal of Defense Software Engineering*, 8(4):11–15, April 1995. Web version, online at [http://www.stsc.hill.af.mil/crosstalk/1995/apr/Lesson s.asp](http://www.stsc.hill.af.mil/crosstalk/1995/apr/Lesson%20s.asp).
- [9] André Sousa, Uirá Kulesza, Andreas Rummler, Nicolas Anquetil, Ralf Mitschke, Ana Moreira, Vasco Amaral, and João Araújo. A model-driven traceability framework to software product line development. In Jon Oldevik, Gøran K. Olsen, Tor Neple, and Richard Paige, editors, *ECMDA Traceability Workshop (ECMDA-TW) 2008 Proceedings*, pages 97–107, Berlin, Germany, 2008. SINTEF ICT.
- [10] A. von Knethen. Change-oriented requirements traceability. support for evolution of embedded systems. In *Software Maintenance, 2002. Proceedings. International Conference on*, pages 482–485. IEEE, 2002.
- [11] Antje von Knethen. A trace model for system requirements changes on embedded systems. In *IWPSE ’01: Proceedings of the 4th International Workshop on Principles of Software Evolution*, pages 17–26. ACM Press, 2001.
- [12] Stefan Winkler and Jens von Pilgrim. A survey of traceability in requirements engineering and model-driven development. *Software and Systems Modeling*, pages 1–37, December 2009.