

Caracterización de los riesgos inherentes a la ingeniería reversa.

Cuevas, Juan Carlos
juancarloscue@gmail.com

Gastañaga, Iris
iris.gastanaga@gmail.com

Gimenez Zens, Inés
igimenezzens@gmail.com

Mana, Franco
manafranco@gmail.com

Universidad Tecnológica Nacional
Facultad Regional Córdoba
Departamento de Ingeniería de Sistemas de Información
Cátedra de Calidad en los Servicios del Software

Maestro López Esq. Cruz Roja Argentina – Ciudad Universitaria
C.P 5016 – Córdoba – Argentina

RESUMEN

Dentro de la línea de investigación de Gestión de la Calidad en Ingeniería Reversa, este artículo hace foco en el proyecto de investigación orientado a caracterizar los riesgos vinculados a la Ingeniería reversa. El mismo tiene como objetivos, además de su caracterización, producir una lista de riesgos a considerar en proyectos de ingeniería reversa como un aporte a la prevención, uno de los principios fundamentales para una gestión más eficiente de la Calidad del Software en aspectos tales como funcionalidad, confiabilidad, usabilidad, mantenibilidad, portabilidad, seguridad, costos y satisfacción de los clientes-usuarios.

Palabras Claves: Riesgos, Ingeniería reversa, Calidad.

CONTEXTO

En oportunidad del WICC 2009, la Cátedra de Calidad en los Servicios del Software de la carrera de Ingeniería en Sistemas de información de la Facultad Regional Córdoba de la Universidad Tecnológica Nacional, presentó una Línea de Investigación orientada a desarrollar un Framework para la Gestión de la Calidad en Ingeniería Reversa.

En el marco de la antes mencionada Línea de Investigación, en el año 2010, se acreditó el primer proyecto de I/D denominado Caracterización de los Riesgos Inherentes a la Ingeniería Reversa ante la

Secretaría de Ciencia y Tecnología dependiente del Rectorado de la Universidad Tecnológica Nacional, el cual fue aprobado en Noviembre de dicho año con su correspondiente financiación por parte de dicha Secretaría.

Este proyecto, tiene por objetivo la identificación de los riesgos inherentes a la Ingeniería Reversa contribuyendo así al avance del conocimiento en la disciplina, como así también de la mencionada asignatura y otras con las que interactúa en la complejidad del currículo de la carrera, de los docentes de la carrera, de los graduados y profesionales del medio y de la Industria del Software.

1 INTRODUCCION

Resumen Técnico

La administración de proyectos de mantenimiento de software es dificultosa. Los inconvenientes se inician con la necesidad de comprender el sistema de software antes de realizar cualquier tipo de cambio (adaptativo, correctivo, preventivo y de perfeccionamiento), lo cual insume una gran porción del tiempo invertido en implementar un cambio. Lo expuesto se fundamenta en cuestiones vinculadas a la documentación, complejidad del producto de software, conocimiento sobre el dominio y del personal. La Ingeniería Reversa

consiste en un proceso que tiene por objetivo obtener información relevante del sistema a partir del código fuente. Este proceso implica el análisis del sistema objetivo, la identificación de sus componentes y sus interrelaciones y la creación de representaciones del sistema en diferentes formas o en elevados niveles de abstracción. A tal fin se realizan abstracciones de funciones, datos y procesos. Para ello sería fundamental llevar este concepto a la práctica mediante la aplicación de un proceso sistemático para facilitar la tarea de comprensión del sistema de software y obtener documentación del mismo que contribuya a eficientizar su mantenimiento.

Por otro lado, como en todo proceso, existe la probabilidad de errores, fallas, etc. los cuales impactan en diferente medida sobre cada uno de los componentes del proceso como así también sobre el proceso considerado como un todo. Esta relación entre probabilidad e impacto nos sitúa en el paradigma de los riesgos. La administración de riesgos está muy ligada a la prevención, elemento este fundamental en la gestión de la calidad.

El abordaje de los riesgos vinculados a la Ingeniería Reversa no es uno de los temas de mayor tratamiento en la bibliografía existente, pero no por ello no es considerado como conveniente y necesario avanzar en su exploración, sobre todo en ambientes de proyectos de software administrados sin la aplicación de métodos ingenieriles y no basados en la filosofía de Calidad.

Estado actual sobre el conocimiento del tema

El Cambio es inherente al software. [Jarzabek:2007]. El mantenimiento de software es la disciplina referida a los cambios relativos a un sistema de software después de su puesta en producción. En los últimos años los productos de software tuvieron un fuerte incremento en cuanto a su cantidad y diversidad de aplicaciones, a tal punto que han invadido en forma creciente nuestra vida cotidiana, y por ende la necesidad de "mantener" dicho software operativo, es decir que tales sistemas cumplan con sus objetivos, en forma eficaz cuando no eficientemente; en definitiva: que sean útiles. Es más, su funcionamiento y uso correcto puede ser una cuestión de vida o muerte. Otra cuestión que no es trivial es que en el ciclo de vida de un software, su

mantenimiento supera en mucho más de lo deseable, el 70% de su ciclo de vida y sus costos totales varían de un 40% a más del 70% en dicho ciclo, lo cual implica la urgente necesidad de encontrar formas de eliminar o reducir los problemas relativos a esta etapa de dicho ciclo.[Grubb-Takang:2005].

Por otro lado, cualquier tipo de cambio debe ser precedido por un proceso de comprensión del sistema de software, el que involucra una gran proporción del tiempo invertido en llevar a cabo el cambio. Entre las principales razones para ello se puede mencionar: documentación no existente, desactualizada o incorrecta, la complejidad del sistema, la indisponibilidad de los desarrolladores originales del sistema software o la carencia del conocimiento suficiente del dominio por parte del encargado de mantenimiento. Una forma de abordar y aliviar estos problemas es la de realizar abstracciones acerca del sistema a partir de información relevante del código fuente. Esto se denomina Ingeniería Reversa. La misma se aplica principalmente a sistemas de software heredados y se fundamenta en abstracciones de datos, procesos y funciones cuya meta es la de facilitar el cambio permitiendo la comprensión de un sistema de software en términos de qué hace, cómo trabaja y la representación de su arquitectura. Los objetivos, al perseguir esta meta, son recuperar información perdida, facilitar migraciones entre plataformas, mejorar y/o proveer nueva documentación, extraer componentes reusables, reducir el esfuerzo de mantenimiento, lidiar con la complejidad y otros. Alcanzar estos objetivos en forma exitosa se traduce en beneficios relacionados al mantenimiento y el aseguramiento de la Calidad. [Grubb-Takang:2005].

Existen diferentes abordajes sobre la Ingeniería Reversa [Grubb-Takang:2005], de código orientado a objetos [Tonella-Potrich:2005] y sus secretos [Eilam:2005], entre otros, con escasa mención y, por ende su profundización, en lo inherente a los riesgos propios de este proceso, limitación ésta que impacta directamente en la Gestión de la Calidad del mismo.

Otros trabajos abordan aspectos que van de los legales a particulares del proceso y herramientas de Ingeniería Reversa, como por ejemplo: August II: Una herramienta para el paso a paso de la Ingeniería Reversa del Modelo de Datos

[Davis:1995], Uso de técnicas formales e informales en Ingeniería Reversa para programas en C [Gannod-Cheng:1996], Comparación de cuatro herramientas de Ingeniería Reversa [Bellay-Gall:1997], Lecciones aprendidas en Ingeniería reversa de Datos [Davis:2001], Herramientas para Ingeniería Reversa como medio para Conocimiento imperfecto [Jahnke-Valenstein:2000], atributos de la calidad – aplicabilidad, extensibilidad y escalabilidad-[Tilley:1998], Métricas para Ingeniería Reversa [Zhou et al:1999], cuestiones de Calidad para Ingeniería Reversa en Base de Datos: el método, adquisición de la semántica del dominio en forma minuciosa, y criterio de evaluación de performance [Chiang-Barron:1995], Hacia Un abordaje efectivo para Ingeniería reversa [Garcia et al:2004], Ingeniería reversa con Acoplamiento Lógico [D'Ambros-Lanza:2006], e Ingeniería de Reversa para Requerimientos destacando aspectos vinculados a simplificar la complejidad y mejora de la calidad del software [Fahmi-Choi:2007], entre otros, con igual carencia de un adecuado abordaje de la temática de riesgos vinculados a la ingeniería reversa.

Asimismo, en lo referente a la administración de riesgos vinculados al software, se puede observar un abordaje fundamentalmente orientado a la producción de software, como por ejemplo B. Boehm que postula que la identificación de riesgos en etapas tempranas del desarrollo disminuye los costos a largo plazo y ayuda a prevenir desastres de software [Boehm:1991], R. Fairley propone un proceso de 7 pasos que describe cómo identificar factores de riesgos, calcular su probabilidad y efecto sobre un proyecto, planear y conducir la administración de riesgos [Fairley:1994], Charette propone que el paradigma de la "ciencia postnormal" ofrece una mejor adaptación mediante el uso de un abordaje de administración robusto basado en una ética de tomar riesgos para grandes proyectos de software [Charette:1996], Boehm y DeMarco plantean que si bien en las disciplinas de ingenierías maduras la administración de riesgos ha sido de rigor por centurias, se preguntan cómo se aplica a los últimos negocios riesgosos: el desarrollo de software [Boehm-DeMarco:1997] a la vez que analizan otros abordajes tales como los de: Lister [Lister:1997], Carr [Carr:1997], Carvey, Phair y Wilson [Carvey et al:1997], Moyniham [Moyniham:1997] y otros; y el trabajo de Charette,

Adams y White que abordan el tema de la Administración de Riesgos en el Mantenimiento de Software [Charette et al:1997] como uno de los pocos trabajos que vinculan el tema de riesgos en la etapa del mantenimiento dentro del ciclo de vida del software. Es decir las publicaciones relativas a riesgos vinculadas al mantenimiento decrecen en cantidad muy considerable respecto al desarrollo del software y son casi nulas en lo que se refiere específicamente a Ingeniería reversa.

Lo anteriormente puntualizado genera la necesidad de profundizar el conocimiento del estado del arte en lo referente a Ingeniería Reversa en general y las prácticas de Gestión de sus Riesgos asociados como contribución a la Calidad en dicho ámbito, en particular.

Metodología

i. Estudio exploratorio de carácter bibliográfico sobre Mantenimiento de Software e Ingeniería Reversa

Se analizarán los trabajos realizados por los siguientes Autores:

- Penny Grubb y Armstrong A. Takang
- Paolo Tonella y Alessandra Potrich.
- Eldad Eilam
- Scot Tilley
- Stanislaw Jarzabek
- Roger H. L. Chiang y Terence M. Barron
- Shikun Zhou, Hongji Yang, Paul Luker y Xudong He
- Otros

El abordaje de estos trabajos estará orientado a conocer el estado del arte actual sobre la gestión del mantenimiento en general y de la Ingeniería Reversa y la gestión del riesgo en forma integrada en particular.

ii. Identificación del universo poblacional objeto de estudio

iii. Desarrollo de un Plan de recopilación de la información

iv. Desarrollo de la herramienta para recopilar la información

v. Estudio descriptivo de la Ingeniería Reversa en la industria del Software de la Ciudad de Córdoba (Polo Industrial de Software de mayor importancia en el contexto Nacional)

- Unidad de Análisis: Productos Software heredados por las Empresas del Polo Industrial de la ciudad de Córdoba.

- Principales variables a estudiar
 - a) Problemas acaecidos durante la ejecución del proceso de Ingeniería Reversa
 - Tipología
 - Frecuencia
 - Impacto
 - b) Resultados obtenidos por nivel de abstracción
 - c) Nivel de vulnerabilidad de las organizaciones responsables del proceso de Ingeniería Reversa.
 - Técnicas
 - a) Entrevistas
 - b) Encuestas
 - c) Observaciones
 - d) Estadísticas
 - vi. Análisis de la información recopilada.
 - vii. Caracterización de riesgos.
- A estos fines se utilizará la metodología propuesta por el análisis modal de fallos y efectos (AMFE), la que involucra los siguientes pasos:
1. Identificar funciones y operaciones del proceso de mantenimiento.
 2. Determinar los modos de fallos (riesgo) potenciales.
 3. Determinar efectos de fallos potenciales.
 4. Determinar causas de fallos potenciales.
 5. Identificar controles actuales.
 6. Estimar probabilidad de ocurrencia de cada fallo.
 7. Estimar gravedad de fallos
 8. Estimar probabilidad de no detección
 9. Calcular el número de prioridad de riesgo.

LINEAS DE INVESTIGACION Y DESARROLLO

El presente proyecto forma parte de la línea de investigación orientada al Desarrollo de un Framework para la Gestión de la Calidad en Ingeniería Reversa, que involucra;

El estado del Arte en Mantenimiento de Software

El estudio del Estado del arte en Ingeniería Reversa

Aplicación de los principios de Gestión de la Calidad a la Ingeniería Reversa

Gestión de Riesgos en la Ingeniería reversa, como herramienta de prevención, aspecto éste de vital

importancia en los sistemas de Gestión de la Calidad.

Investigación y desarrollo de buenas prácticas para Ingeniería Reversa

Desarrollo de un Marco de Trabajo para la gestión de la calidad en Ingeniería reversa

RESULTADOS OBTENIDOS/ESPERADOS

Identificar los riesgos inherentes a la Ingeniería Reversa como herramienta que contribuya a la productividad, eficiencia y Calidad en su gestión

Producir una Lista de los Riesgos Caracterizados vinculados a la Ingeniería reversa.

Aportar a una gestión más eficiente del software en aspectos tales como funcionalidad, confiabilidad, usabilidad, mantenibilidad, portabilidad, seguridad, costos y satisfacción de los clientes-usuarios.

Transferencia de los resultados obtenidos en materia de investigación y desarrollo a Empresas vinculadas a la industria del software a nivel local, nacional e internacional, mediante cursos, talleres, seminarios, y publicaciones en congresos y revistas especializadas.

FORMACION DE RECURSOS HUMANOS

Los resultados obtenidos estarán destinados a:

Los integrantes del Grupo de Investigación

Docentes de la Cátedra

Docentes de otras asignaturas

Estudiantes

Graduados

Especializandos, Mastrandos y Doctorandos.

Profesionales de la Industria del Software

Produciendo aportes en aspectos tales como:

Incremento de competencias profesionales

Incremento de conocimientos científicos, ingenieriles y administrativos en materia de

Riesgos caracterizados

Estado del arte

Herramientas

Metodologías
Mejora continua de procesos

BIBLIOGRAFIA

[Jarzabek:2007] Stanislaw Jarzabek; "Effective Software Maintenance and Evolution. A Reuse-Based Approach". Auerbach Publications. NY - EEUU - 2007.

[Grubb-Takang:2005] Penny Grubb and Armstrong A. Takang; "Software Maintenance - Concepts and Practice" Second edition. World Scientific - Singapore - 2005.

[Tonella-Potrich:2005] Paolo Tonella and Alessandra Potrich; "Reverse Engineering of Object Oriented Code". Springer Science + Business Media, Inc. - Boston - EEUU - 2005

[Eilam:2005] Eldad Eilam "Reversing - Secrets of Reverse Engineering". Wiley Publishing, Inc - IN - EEUU - 2005.

[Davis:2005] Kathi Hogshead Davis " August-11: A Tool for Step-by-Step Data Model Reverse Engineering" IEEE - 0-8186-7111-4/95 - 1995

[Gannod-Cheng:1996] Gerald C. Gannod and Betty H. C. Cheng " Using Informal and Formal Techniques for the Reverse Engineering of C Programs" - 0-8186-7674-4/9 - IEEE - 1996.

[Bellay-Gall: 1997] Berndt Bellay and Harald Gall; "A Comparison of four Reverse Engineering Tools" 0-8186-81624/97 - IEEE - 1997.

[Davis:2001] Kathi Hogshead Davis; "Lessons Learned in Data Reverse Engineering". 1095-1350/01 - IEEE - 2001.

[Jahnke-Valenstein:2000] Jens H. Jahnke and Andrew Walenstein "Reverse Engineering Tools as Media for Imperfect Knowledge" 1095-1350/00 - IEEE - 2000.

[Tilley:1998] Scott Tilley; "A Reverse-Engineering Environment Framework". TECHNICAL REPORT

CMU/SEI-98-TR-005. Carnegie Mellon University - Software Engineering Institute - PA - EEUU. 1998.

[Zhou et al:1999] Shikun Zhou, Hongji Yang, Paul Luker and Xudong He; "A Useful Approach to Developing Reverse Engineering Metrics". 0-7695-0368-3/99 - IEEE - 1999.

[Chiang-Barron:1995] Roger H. L. Chiang and Terence M. Barron "Quality Issues in Database Reverse Engineering: An Overview". 95 ENGINEERING MANAGEMENT CONFERENCE - 0-7803-2799-3/95/ - IEEE - 1995.

[Garcia et al:2004] Vinicius C. Garcia, Daniel Luc'edio, Antonio F. do Prado, Alexandre Alvaro and Eduardo S. de Almeida "Towards an effective approach for Reverse Engineering". Proceedings of the 11th Working Conference on Reverse Engineering (WCRE'04) - 1095-1350/04 - IEEE - Computer Society - 2004.

[D'Ambros-Lanza:2006] Marco D'Ambros and Michele Lanza "Reverse Engineering with Logical Coupling". Proceedings of the

13th Working Conference on Reverse Engineering (WCRE'06) - 0-7695-2719-1/06 - IEEE - Computer Society - 2006.

[Fahmi-Choi:2007] Syed Ahsan Fahmi and Ho-Jin Choi; "Software Reverse Engineering to Requirements". 2007 International Conference on Convergence Information Technology - 0-7695-3038-9/07 - IEEE - Computer Society - 2007.

[Boehm:1991] Barry W. Boehm "Software Risk Management: Principles and Practice" 0740-7459/91/0100/0032. IEEE. 1991.

[Fairley:1994] Richard Fairley "Risk Management for Software Projects". 0740-7459/94. IEEE. 1994.

[Crarett:1996] Robert N. Charette "Large-Scale Project Management is Risk Management". 0740-7459/96. IEEE. 1996

[Boehm-De Marco:1997] Barry W. Boehm and Tom DeMarco " Software Risk Management". 0740-7459/97. IEEE. 1997.

[Lister:1997] Tim Lister "Risk Management is Project Management for Adults". IEEE Software - May-Jun/1997.

[Carr:1997] Marvin J. Carr "Risk Management May Not Be for Everyone". IEEE Software. May-Jun/1997.

[Carvey et al:1997] Paul R. Carvey, Douglas J. Phair and John A. Wilson " An Information Architecture for Risk Assessment and Management". IEEE Software. May-Jun/1997.

[Moynihan:1997] Tony Moynihan "How Experienced Project Managers assess Risk". IEEE Software. 1997.

[Charette et al:1997] Robert N. Charette, Kevin MacG. Adams and Mary B. White " Managing Risk in Software Maintenance". IEEE Software. May-Jun/1997