

# Herramienta para la Gestión de Riesgos en Proyectos de Software

Rodolfo Bertone<sup>1</sup>, Pablo Thomas<sup>2</sup>, Daniel Taquias<sup>3</sup>, Sebastián Pardo<sup>4</sup>

Instituto de Investigación en Informática – III LIDI<sup>5</sup>  
Facultad de Informática - UNLP

**Resumen.** El propósito de este trabajo es presentar el desarrollo de una herramienta de software orientada PyMEs, para administrar la gestión de riesgos en proyectos de desarrollo de Sistemas Informáticos. La herramienta permite gestionar el conocimiento generado de manera de enriquecer el análisis de proyectos subsiguientes. Se gestionan los riesgos de un proyecto desde la etapa de elicitación de requerimientos y durante todo el ciclo de vida. De esta forma se puede administrar de manera proactiva cada acontecimiento presentado durante el análisis, diseño, codificación, prueba y mantenimiento del software, y además es posible disponer de suficiente material para generar planes alternativos de contingencia. La herramienta Web creada es de uso libre y brinda la posibilidad de efectuar tareas de identificación de riesgos, planes de riesgos en base a taxonomías estándar, y planes de contingencia asociados a cada riesgo.

**Keywords:** Ingeniería de Software, Riesgos, Planificación de Riesgos, Herramienta CASE

---

1 pbertone@lidi.info.unlp.edu.ar Profesor Titular Dedicación Exclusiva, Facultad de Informática UNLP  
2 pthomas@lidi.info.unlp.edu.ar Profesor Adjunto Dedicación Exclusiva, Facultad de Informática UNLP  
3 danieltaquias@hotmail.com Licenciado en Informática, Facultad de Ingeniería, UNPSJB  
4 spardoarg@yahoo.com.ar Analista de Computación, Facultad de Informática, UNLP  
5 50 y 120, 2do piso, La Plata, 1900, Argentina, +54 221 4227707

## 1 Introducción

En los proyectos de software, la gestión de riesgos resulta un escalón vital para obtener el éxito en el desarrollo de tales proyectos. Las presiones de la competencia, los cambios normativos y la evolución de las técnicas, pueden obligar a los jefes de proyectos de software a alterar los planes y estrategias durante la ejecución de un proyecto. Los cambios en los requerimientos de los usuarios, las nuevas herramientas y tecnologías, las constantes amenazas de seguridad y los cambios en la plantilla de empleados, añaden más presión al equipo de proyectos de software y perjudican la toma de decisiones [1].

No son muchas las organizaciones que utilizan métodos de gestión de riesgos en sus proyectos de desarrollo de software. Estudios realizados demuestran esta problemática claramente; por ejemplo, durante el año 2001, KLCI (2001) realizó y publicó un estudio con 268 organizaciones de todo el mundo, referente a la aplicación de la gestión de riesgos, y el resultado fue que el 3% no utilizaba ningún marco de gestión del riesgo, el 18% utilizaba un marco propio para identificar sus riesgos, el 37% de los participantes habían utilizado algún marco informal, el 28% utilizaban procedimientos repetitivos, y sólo un 14% utilizaba un enfoque formal para identificar riesgos. Dicho estudio permitió concluir que, aunque algunas organizaciones usen procesos formales de gestión del riesgo en otras partes de su negocio, demuestran una gestión de riesgos pobre en el ámbito general de los sistemas de información; utilizando algún enfoque informal de gestión de riesgos o ignorando los posibles riesgos a desarrollarse en los avances de proyectos [2].

Actualmente las pequeñas organizaciones de desarrollo de software, de reciente entrada al mercado, no utilizan procesos formales de gestión de riesgos, pues se conforman solamente con definir enfoques informales, debido a factores como: falta de conocimiento sobre posibles métodos y herramientas, limitaciones prácticas y teóricas de los marcos de gestión de riesgos que entorpecen la facilidad de uso de éstos métodos, además de la ausencia de estudios reales con evaluaciones sistemáticas o científicas.

Administrar los riesgos por medio de un proceso de desarrollo de software predecible, provee un fundamento que permite generar en forma consistente el software para el usuario final, y a un menor costo. A partir de esta base, se podrán adoptar técnicas y herramientas para lograr que los desarrolladores sean más productivos, para elevar la calidad del software, y para automatizar muchos de los procesos de gestión del software, liberando de esta manera más tiempo para el desarrollo de las aplicaciones. [3]

Por lo tanto, el propósito de este trabajo fue aplicar los conceptos de la gestión de proyectos y la gestión de riesgos en proyectos de desarrollo de software, implementando un proceso proactivo de administración de riesgos, a lo largo del ciclo de vida del proyecto.

El SEI define riesgo como “la posibilidad de sufrir una pérdida” [4] y a la Gestión de Riesgos como “la práctica compuesta de procesos, métodos y herramientas que

posibilita la gestión de los riesgos en un proyecto y que provee de un entorno disciplinado para la toma de decisiones proactiva en base a determinar constantemente que puede ir mal, e identificar cuáles son los riesgos más importantes, enfocarse e implementar estrategias para gestionarlos”; esta actividad se inicia en la primera etapa de un proyecto de software (durante la exploración de conceptos) y se desarrolla a lo largo de todo su ciclo de vida (hasta la aceptación del producto del proyecto) [5].

La gestión de riesgos es importante debido a que ayuda a evitar desastres, re-trabajo y sobre-trabajo, pero aún mas importante, porque estimula la generación de situaciones del tipo ganar-ganar (win-win) [6].

Una adecuada Gestión de Riesgos permite el óptimo aprovechamiento de recursos y provoca, como consecuencia, el aumento de ganancias y la disminución de pérdidas. La ausencia de una apropiada Gestión de Riesgos conlleva a la imposibilidad de lograr el control efectivo de un proyecto, derivando esto en la imposibilidad de realizar una correcta gestión. En consecuencia, la Gestión de Riesgos debe ser enfatizada y considerada como una actividad clave en todo tipo de proyectos, particularmente en proyectos de desarrollo de software [8] [9].

Al realizar Gestión de Riesgos, es fundamental lograr una descripción clara del riesgo de forma tal de que éste pueda ser comprendido y administrado adecuadamente [7]; cuando se lo enuncia, no sólo debe considerarse el síntoma sino también sus consecuencias. Existen varios modelos de Gestión de Riesgos, pero el más aceptado consta de cinco pasos secuenciales e iterativos [3]. Estos pasos son Identificación, Análisis, Planificación, Seguimiento y Control. Paralelamente a éstos pasos existen dos actividades comunes a ellos: documentación y comunicación [10].

La Identificación de Riesgos en proyectos de software consiste en la determinación de elementos de riesgos potenciales mediante la utilización de algún método consistente y estructurado; este es el paso más importante entre todos aquellos que componen las actividades de Gestión de Riesgos, ya que sin la correcta determinación de los riesgos, no es posible desarrollar e implementar anticipadamente respuestas apropiadas a los problemas que puedan surgir en el proyecto [11]. El resultado de la identificación de riesgos es una lista que contiene los riesgos y su categoría correspondiente.

El diccionario Webster [12] define a las Taxonomías como las clasificaciones ordenadas de plantas y animales de acuerdo a sus relaciones naturales presumidas; esta definición suele ser ampliada especificando a las Taxonomías como las clasificaciones ordenadas de elementos de acuerdo a sus relaciones presumidas; así consideradas, puede emplearse a las taxonomías como herramientas de suma utilidad en diferentes ramas de la ciencia y de la industria, donde se pretende organizar y facilitar el acceso a un número importante de elementos que se encuentran mutuamente relacionados de alguna manera relevante.

La Gestión de Riesgos en base a Taxonomías implica utilizar una estructura de clasificación, como una lista de consulta durante la actividad de Identificación de los Riesgos [13].

Adicionalmente CMMI [14], estándar a nivel mundial para la medición de la calidad de los procesos de desarrollo de software, presenta la Gestión de Riesgos como una de sus áreas de procesos fundamentales de Nivel 3.

Por lo tanto, los aspectos mencionados son claves en la Administración de Proyectos de Software, y existe una creciente necesidad en las compañías de desarrollo de software, de disponer herramientas que permitan automatizar y estandarizar sus procesos de gestión, en busca de una mayor madurez organizacional.

## **2 Gestión de Riesgos**

Un proceso efectivo de gestión de riesgos es un componente importante en todo proyecto de desarrollo de software exitoso. La gestión de riesgos permite definir en forma estructurada, operacional y organizacional, una serie de actividades para gestionar los riesgos de los proyectos a lo largo de todas las fases de su ciclo de vida de desarrollo de software. En la mayor parte de los casos, esto se traduce en la creación de planes tendientes a impedir que los riesgos se transformen en problemas, o a minimizar su probabilidad de ocurrencia o impacto.

A nivel organizacional, una política de gestión de riesgos debería considerar, al menos, los siguientes aspectos [15]:

- Todos los riesgos relacionados con los proyectos de desarrollo de software deben ser identificados, analizados, priorizados y revisados, siguiendo un Plan de Gestión de Riesgos.
- El Plan de Gestión de Riesgos debe ser documentado apropiadamente, ya sea como un documento independiente o formando parte del Plan de Proyecto. Como consecuencia de los constantes cambios, la lista de los riesgos y la información relacionada con su estado actual e historia reciente, deben ser gestionados en una Base de Datos específica de Riesgos.
- La información contenida en la Base de Datos de Riesgos debe ser enriquecida a través del tiempo.

En [16] se desarrollan en detalle los aspectos considerados para el desarrollo de la herramienta presentada en este trabajo.

### **2.1. Modelo de Gestión de Riesgos**

El modelo de gestión de riesgos seleccionado consta de cinco pasos, etapas o fases: Identificación, Análisis, Planificación, Seguimiento y Control, compartiendo como actividades comunes la Documentación y Comunicación.

La primera actividad consiste en la identificación del riesgo, esta actividad debe llevarse a cabo de forma disciplinada y consistente, alentando a cada uno de los miembros del equipo de proyecto a expresar sus preocupaciones y posibilitando de este modo su análisis posterior.

Durante la etapa de análisis, se pretende transformar los datos obtenidos en información que permita toma de decisiones enfocada en los riesgos más importantes para el proyecto. El análisis de riesgos es un proceso sistemático de estimación de la probabilidad de ocurrencia, y la magnitud de la pérdida o impacto de cada uno de los riesgos identificados. De este modo se logra reducir la incertidumbre de la medida y del resultado del acontecimiento asociado a cada riesgo.

En la fase de planificación se transforma a la información relacionada con los riesgos, en decisiones y acciones presentes y futuras. La planificación involucra tanto el desarrollo de acciones individuales para cada uno de los riesgos, como la priorización de dichas acciones y la creación de un plan integral de administración.

El seguimiento consiste en monitorear constantemente tanto el estado de los riesgos como las acciones tomadas, con el propósito de evitarlos o mitigar su impacto negativo. El seguimiento de los riesgos trae inevitablemente asociada la toma de una serie de métricas relacionadas con la gestión de riesgos, que permitan a los responsables de la administración del proyecto llevar a cabo una estricta y permanente verificación de los planes.

El control de riesgos posibilita corregir las desviaciones que puedan producirse en los planes de acción implementados.

La comunicación y documentación del proceso son claves en el modelo. La comunicación está presente y se manifiesta a diferentes niveles en el modelo: un primer nivel lo constituye la comunicación que debe realizarse entre los diferentes pasos del proceso, un segundo nivel está caracterizado por la comunicación interna del equipo de proyecto, y un tercer nivel por la que surge entre el proyecto y todos los stakeholders.

## **2.2. Estrategias para Tratar Riesgos**

Una de las más importantes decisiones tiene que ver con la estrategia de respuesta a adoptar para cada uno de los riesgos sobre los cuales se focaliza la atención del proyecto. Formalmente, se define a una estrategia de respuesta como una actividad que puede ser realizada con el propósito de manejar un riesgo antes de que el mismo se dispare. Es importante considerar que para cada uno de los riesgos identificados puede seleccionarse una estrategia diferente, y que la estrategia elegida puede ser reemplazada a medida que se producen revisiones sucesivas de los riesgos.

Existen diferentes estrategias de respuesta, entre las más reconocidas y utilizadas pueden destacarse: *aceptar el riesgo* y su dificultad o imposibilidad de control; *evitar el riesgo*, generando cambios en las condiciones iniciales del problema; *controlar el riesgo*, tomando acciones concretas; *investigar las causas y posibles soluciones*; *reducir el efecto del riesgo*, analizando acciones plausibles; y *transferir el riesgo* hacia otro actor del problema.

## **3. Marco teórico de la Propuesta**

La herramienta de software propuesta sigue el lineamiento de un producto tipo CASE (Computer Aided Software Engineering – Ingeniería de Software Asistida Por Computadora), que permite la administración integral de riesgos en un proyecto de software, y además posibilita generar una base de conocimiento para mejorar el tratamiento de riesgos en proyectos futuros.

El desarrollo realizado permite generar las condiciones de trabajo necesarias para identificar, analizar, planificar, realizar el seguimiento y el control de riesgos, en un proyecto de software desde su comienzo y mientras éste se encuentre activo. De esta

forma se puede tratar de manera proactiva cada situación de riesgo presentada, o en su defecto disponer con suficiente material para generar planes alternativos de contingencia.

Una vez finalizada la etapa de construcción de un sistema de Información, luego que el sistema comienza a funcionar, es posible, generar un análisis post mortem del proyecto de desarrollo. Este análisis compara los resultados previstos con los resultados obtenidos. Así, es posible registrar las desviaciones producidas e indicar su fuente de origen.

De esta forma, es posible generar una base de conocimiento de riesgos dentro de la organización que desarrolla software. Esta base de conocimiento permite documentar las mejores prácticas de casos de riesgo y el contexto en el cual fueron llevadas a cabo; éstas son lecciones aprendidas para el análisis de viabilidad de nuevos proyectos.

#### **4. Herramienta de Gestión de Riesgos**

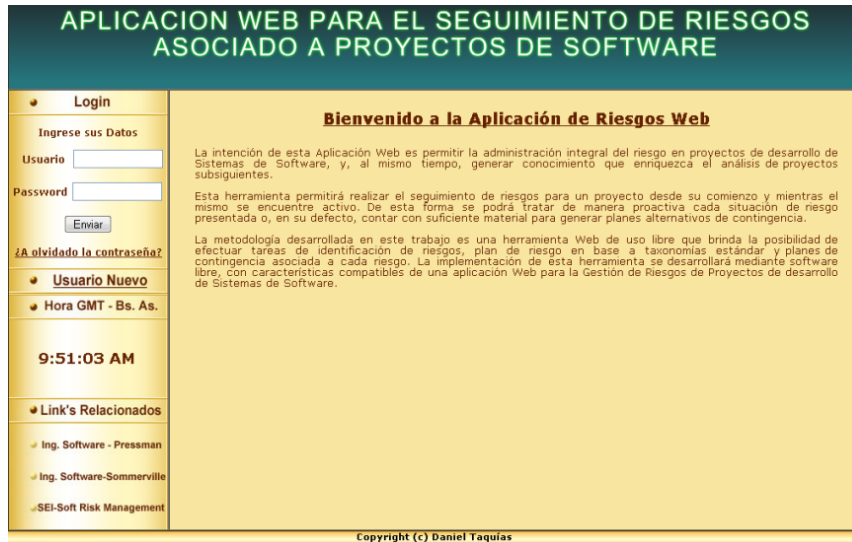
Los objetivos principales en el desarrollo de esta aplicación son:

- Poder definir los casos de riesgos que con mayor frecuencia se presenta en un proyecto de software. En este caso, la herramienta permite al responsable de riesgos del proyecto, obtener información sobre los riesgos de otros proyectos relacionados, con fines comparativos y ejemplificativos.
- Estudiar alternativas de tratamiento de los riesgos, tanto de los planes de contingencia como los planes para el seguimiento de las situaciones de riesgo, basándose para ello en la información histórica disponible.
- Generar el documento de Reducción, Supervisión y Gestión del Riesgo que permita su administración integral en proyectos de desarrollo de sistemas de software.
- Generar el documento de Reducción, Supervisión y Gestión del Riesgo Post Mortem, para comparar los resultados previstos con los resultados obtenidos, registrando las desviaciones que se podrían haber producido y la fuente de origen.
- Detectar los riesgos que más frecuentemente aparecen, y los que no fueron previstos.

Todo esto posibilita administrar de una forma ordenada, concreta y con un cierto nivel de precisión, el plan de gestión de riesgos de una organización.

La herramienta fue desarrollada utilizando software de uso libre. El desarrollo se realizó en PHP, con MySQL y Apache.

La figura 1 presenta la interfaz principal de la herramienta.



**Figura 1.** Acceso a la herramienta.

Las opciones de menú disponibles, una vez conectado al sistema, permitirán administrar cada uno de los proyectos de software que dispone la organización, tal como se muestra en la figura 2.

**Figura 2.** Definición de un Proyecto.

En cada proyecto es posible definir y controlar el plan de riesgo asociado, de acuerdo a los conceptos previamente mencionados y con la interface que se presenta en la figura 3.

Los riesgos definidos serán volcados en una Tabla de Riesgos. En ella se presentan todos los riesgos ingresados para el proyecto en cuestión. Como cada riesgo fue definido con su impacto y probabilidad de ocurrencia, la tabla de Riesgos presentada establece un criterio posible de ordenamiento donde se pondera por igual

tanto el impacto como la probabilidad de ocurrencia. Dentro de proceso de análisis de riesgo es posible alterar el orden de tratamiento de los riesgos en caso que el usuario así lo considere.

The screenshot shows a web application interface for risk management. At the top, it says 'Proyecto de Software Empresa ROBERTO S.R.L.' and 'Carga de Riesgos'. The form includes several fields: 'Código de Riesgo' with the value '1', 'Categoría del Riesgo' with a dropdown menu set to 'Tecnológico', 'Nombre del Riesgo' with an empty text box, 'Tipo de Riesgo' with an empty text box, 'Probabilidad del Riesgo' with a dropdown menu set to '05%', and 'Impacto del Riesgo' with a dropdown menu set to 'Despreciable'. Below these are four larger text input areas for 'Descripción del Riesgo', 'Factores que influyen:', 'Reducción del Riesgo:', and 'Supervisión del Riesgo:'.

**Figura 3.** Definición de un riesgo en un proyecto.

Por último, es responsabilidad del analista de riesgos definir la ubicación de la línea de corte. Esta línea, tal como se presenta en la figura 4, establece la división entre los riesgos que serán tratados en el proyecto y los que serán excluidos. Este paso no debe soslayarse. Todos los riesgos que queden por encima de la línea de corte deberán ser analizados en detalle, y poseer un plan de contingencia. Los riesgos descartados en caso de presentarse deberán ser tratados reactivamente.

La herramienta generada también permite analizar los riesgos Post Mortem al proyecto. De esta forma es posible documentar todas las desviaciones producidas, con la finalidad de generar una base de conocimiento que enriquezca futuros trabajos de la organización. Además, los planes de contingencia previstos pueden ser cotejados contra la realidad, permitiendo indicar fortalezas y debilidades encontradas para refinar planes de riesgos en proyectos futuros.

Finalmente, la herramienta Web desarrollada brinda la posibilidad de efectuar la administración de riesgos, en base a taxonomías estándares para la implementación de planes de Reducción, Supervisión y Gestión del Riesgo. La herramienta generada puede encontrarse en el sitio <http://www.riesgosweb.info/>.



Proyecto de Software Mueblería 'El Mueble Feliz'			
Línea de Corte			
Número de Riesgo	Riesgo	Probabilidad	Impacto
1	Plan de prueba ineficiente por falta de experiencia del personal	65 %	Catastrófico
2	Medidas inadecuadas por falta de experiencia del personal	65 %	Crítico
3	Escasa comunicación con el cliente	60 %	Crítico
4	Calendarios no cumplidos	55 %	Crítico
LÍNEA DE CORTE - LÍNEA DE CORTE			
5	Análisis de riesgos inadecuado	45 %	Catastrófico
6	Seguimiento del proyecto y documentación incorrectos	40 %	Crítico
7	Cambios de requerimientos del Usuario	35 %	Crítico

**Figura 4.** Definición de línea de corte.

## 5 Conclusiones

Los autores de este trabajo han realizado estudios, en los cuales se ha detectado que uno de los problemas fundamentales en el aseguramiento de calidad en las PyMEs productoras de software del país, se origina en la Administración de Riesgos. Este punto es, o bien no tenido en cuenta, o bien analizado de manera parcial debido a: falta de tiempo, trabajar de manera reactiva respondiendo espontáneamente a un caso de riesgo, y además por no contar con herramientas útiles y de bajo costo o libres, que permitan generar un Plan de Riesgo.

En España por ejemplo, se ha trabajado en el mismo sentido [17]; pero en general, los productos no son de uso libre y por ende es difícil su evaluación y comparación con la herramienta presentada. Además, no es fácil sensibilizar a las PyMEs para que inviertan en productos con las características enunciadas en este trabajo.

La herramienta generada intenta abarcar los aspectos más relevantes de la Gestión de Riesgos. Permite administrar el Plan de Riesgo Integral de una organización a partir de la gestión de riesgos en cada proyecto involucrado. En consecuencia, este producto puede ser de gran utilidad para cualquier PyME que intente mejorar su Administración de Riesgos.

## 6 Trabajos Futuros

El primer objetivo consiste en mejorar la interfaz de usuario, con el fin de lograr mayor usabilidad.

El segundo propósito es poner en funcionamiento el producto en un entorno de prueba real, más allá de los entornos experimentales generados. Este trabajo fue

desarrollado como parte de tesis de grado y ha sido evaluado en entornos de simulación. Actualmente existen PyMEs que están comenzando a utilizar el producto, por lo que se prevé nuevos requerimientos para incrementar las prestaciones que brinda la herramienta desarrollada.

## 7 Referencias

1. Ingeniería de Software, Ian Sommerville, Pearson, Addison Wesley, 2006
2. Software Risk Management Practices. KLCL research group report. 2001
3. Ingeniería de Software. Un enfoque Práctico. Roger Pressman. McGraw Hill. 2008
4. Software Engineering Institute. Disponible en <http://www.sei.cmu.edu/risk/>
5. Ingeniería de Software, Shari Pfleger, Prentice Hall, 2008
6. Continuous Risk Management at NASA. Rosenberg, L., Hammer, T., Gallo, A., [http://satc.gsfc.nasa.gov/support/ASM\\_FEB99/crm\\_at\\_nasa.html](http://satc.gsfc.nasa.gov/support/ASM_FEB99/crm_at_nasa.html)
7. A spiral model of software development and enhancement. B. Boehm, IEEE Computer, 21(5), 61-72. 1988
8. Managing Risk: Methods for Software Systems Development. E. Hall, Reading, MA: Addison Wesley, 1998
9. Managing software Quality and Business Risk. M. Ould. John Wiley & Son. 1999
10. Software Risk Management, S. Zardari. Information Management and Engineering, 2009. ICIME '09. International Conference on Digital Object Identifier: 10.1109/ICIME.2009.138, 375-379, 2009.
11. Quality Software Project Management. Futrell, Shafer & Shafer. Editorial Prentice Hall. ISBN 0130912972. 2002
12. Merriam-Webster's Collegiate Dictionary. Disponible en <http://www.webster.com> . 2004
13. Taxonomy-Based Risk Identification. Marvin J. Carr et al. Disponible en [www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf](http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr06.93.pdf), 1993
14. Capability Maturity Model Integration. Software Engineering Institute. Disponible en <http://www.sei.cmu.edu/cmmi/index.cfm> 2002
15. Motorola Land Mobile Products Sector, "Risk Management". Motorola LMPS, 1999
16. Desarrollo de una Herramienta CASE para el seguimiento de Riesgos Asociados a Proyectos de Software. Daniel Taquias. Tesina de la licenciatura en Informática. UNPSJB, sede Comodoro Rivadavia, 2010
17. (<http://www.csi.map.es/csi/pg5m20.htm>)