Revista del Instituto de Investigaciones FIGMMG Vol. 10, Nº 19, 55-61 (2007) UNMSM ISSN: 1561-0888 (impreso) / 1628-8097 (electrónico)

Gestión de riesgos con CMMI, RUP e ISO en Ingeniería de Software Minero

Gestion de riesgos con CMMI, RUP e ISO en Ingenieria de Software Minero

Alfonso Romero B.[1], Daniel Lovera D.[1], Simeón Yaringaño Y.[1], Silvana Flores Ch.[2]

RESUMEN

Los periodos de cambio y evolución continua de las tecnologías de información y comunicación, cada vez más se ven reducidos en tiempo, así tenemos; hoy que en materia de software o programas informáticos de aplicación en minería; estos entran en estado de obsolescenci a en un promedio de doce meses desde su puesta en funcionamiento. En los últimos dos años el diseño de nuevas aplicaciones informáticas han sufrido cambios importantísimos en el enfoque inicial para su diseño, análisis y codificación, así pues; desde que Bohem en 1982 advirtió algunas técnicas relacionadas con la ingeniería de software hoy observamos que estas técnicas y metodologías de análisis y diseño de aplicaciones informáticas han variado significativamente en relación al enfoque inicial.

La continúa aplicación de las normas y técnicas como CMMI, RUP e ISO han hecho que estos sean perfeccionados año a año, apareciendo de esta manera las versiones beta y/o versiones en general.

En este articulo mostramos que el punto critico de la elaboración y creación de aplicaciones informáticas radica en la gestión de riesgos del proyecto de ingeniería de software, así pues; cada técnica o metodología como el CMMI, RUP o ISO tienen siempre enfocado en su contenido la gestión de riesgos en proyectos de ingeniería de los programas de aplicación en minería.

Palabras clave: Software minero, Proyectos de software minero, riesgos en software minero.

ABSTRACT

The periods of change and continuous evolution of the information technologies and communication, every time has been reduced, thus we have; today that in the matter of software or computer science programs of application; these enter state of obsolescence in an average of six months from their put into operation. In the last two years the design of new computer science applications have undergone the most important changes in the initial approach for their design, analysis and codification, therefore; ever since Bohem in 1982 warned some techniques related to the software engineering today we observed that these techniques and methodologies of analysis and design of computer science applications have varied significantly in relation to the initial approach.

The continuous application of the norms and techniques like CMMI, RUP and ISO have done that these are perfected year to year, appearing of this way the beta versions and/or versions in general.

In this article we showed that the point I criticize of the elaboration and creation of computer science applications is in the management of risks of the project of software engineering, therefore; each technique or methodology like the CMMI, RUP or ISO always have focused in their content the management of risks in projects of software engineering.

Keywords: Mining software, projects of mining software, risks in mining software.

1 Docentes de la Facultad deIngeniería Geológica, Minera, Metalúrgica y Geográfica de la Universidad Nacional Mayor de San Marcos.
2 Estudiante de Maestría de la UNMSM.

RIESGOS EN PROYECTOS DE SOFTWARE

Los proyectos de software para aplicaciones en minería son claramente difíciles de administrar y una gran cantidad de ellos terminan en fracaso. En un proyecto de software minero, éste se puede traducir en una mala calidad del producto, incumplimiento de planes u objetivos y hasta el fracaso del proyecto. La gestión de riesgos en proyectos de software pretende identificar, estudiar y eliminar las fuentes de riesgo antes de que comiencen a amenazar el éxito o la finalización exitosa de un proyecto de desarrollo de software.

Se define el riesgo como la posibilidad que un evento adverso, desgracia o contratiempo pueda manifestarse produciendo una pérdida (Pressman, R., 2001). El riesgo es una posibilidad futura, por lo tanto una gestión adecuada puede determinar la ocurrencia o no ocurrencia de éstos.

Estudios previos han identificado siete categorías de riesgo en proyectos de software, incluyendo:

(1) Gestión,
(2) Clientes y usuarios,
(3) Requerimientos,
(4) Estimación y programación de actividades,
(5) Jefe de proyecto,
(6) Proceso de desarrollo de software y
(7) Personal de desarrollo

COMPONENTES DE LA GERENCIA DE RIESGOS

Se clasifican en dos partes, en variables y metodología de la gerencia de riesgos.

1. Variables de la gerencia de riesgos

Impacto
Probabilidad
Exposición

2. Metodología de la gerencia de riesgos

Planificación de la gerencia de riesgos
Oportunidad de realización
Lanzamientos de la gerencia de riesgos

Análisis de riesgos
Levantamiento de la información
Identificación de los componentes a proteger
Identificación de los riesgos
Priorización de los riesgos

Planificación de la respuesta a los riesgos
Identificación de los planes de contingencia
Evaluación de la efectividad
Plan de implantación
Integración de resultados

Monitoreo y control
Revisión del plan de riesgo
Revisión periódica del grado de implantación de contingencias
Actualizar periódicamente la situación de las variables de riesgo
Planes de emergencia
Acciones correctivas
Lecciones aprendidas

GESTIÓN DE RIESGOS EN INGENIERÍA DE SOFTWARE

Peter Drucker dijo una vez: "Mientras que es inútil intentar eliminar el riesgo y cuestionable el poder minimizarlo, es esencial que los riesgos que se tomen sean los riesgos adecuados". Antes de poder identificar los "riesgos adecuados" que se pueden tomar en un proyecto de software, es importante poder identificar todos los riesgos que sean obvios a jefes de proyectos y profesionales del software.

Riesgos del software

Además de los riesgos técnicos y los de negocio tenemos:

- *Incertidumbre:* El acontecimiento que caracteriza al riesgo puede o no puede ocurrir; por ejemplo, no hay riesgos de un 100 por ciento de probabilidad.
- *Pérdida:* Si el riesgo se convierte en una realidad, ocurrirán consecuencias no deseadas o pérdidas.

Incertidumbre (probabilidad de que ocurra)
Pérdidas
- Producto (rendimiento, mantenimibilidad)
- Proceso de producción (tiempo de desarrollo, coste).

Riesgos del proyecto

Incremento en costes
Desbordamiento organizativo

Riesgos técnicos

Riesgos del negocio

De mercado
De estrategia
De ventas
De gestión
De presupuesto

Estrategias frente al riesgo

Método

- Evaluación previa y sistemática de riesgos.
- Evaluación de consecuencias.
- Plan de evitación y minimalización de consecuencias.
- Plan de contingencias.

Consecuencias

- Evasión del riesgo.
- Menor tiempo de reacción.
- Justificación frente a los superiores.

IDENTIFICACIÓN DE RIESGOS DE PROYECTOS DE SOFTWARE

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, probablemente, el paso más importante entre todos aquellos que componen las actividades de Administración de Riesgos, ya que sin la correcta determinación de los mismos, no es posible desarrollar e implementar anticipadamente respuestas apropiadas a los problemas que puedan surgir en el proyecto [Futrell, A. 2002]. El resultado de la identificación de riesgos es una lista conteniendo los riesgos que se han identificados y su categoría correspondiente.

Existen varios modelos de Administración de Riesgos pero el más aceptado consta de cinco pasos (Identificación, Análisis, Planificación, Seguimiento y Control) los que comparten como actividades comunes las de documentación y comunicación (véase Figura 1).

1. Resumen seguido de su correspondiente traducción al inglés Introducción
2. Texto
3. Conclusiones
4. Agradecimientos
5. Referencias
6. Apéndices (si es aplicable)

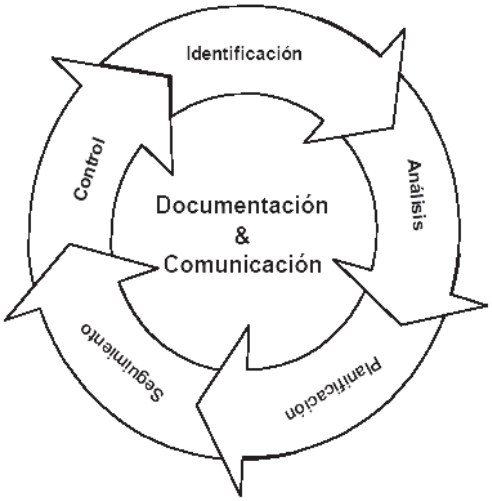

Fig. 1. Modelo de administración de riesgos.

Se recomienda usar el Sistema Internacional de Unidades (SI). El estilo aconsejado contempla primero las unidades métricas seguidas de las unidades inglesas en paréntesis.

GESTIÓN DE RIESGOS EN CMMI

El CMMI (Capability Maturity Model Integrated) [CMMI, 2002] se ha convertido en el nuevo estándar a nivel mundial para la medición de la calidad de los procesos de desarrollo de software y presenta como una de sus PA (Process Area) fundamentales de Nivel 3 la Administración de Riesgos.

Dentro del antes mencionado contexto de riesgos y la Identificación juegan un papel fundamental entre los objetivos planteados para el área de proceso asociada al manejo de riegos debido a que las tareas antes indicadas son consideradas como Actividades en el referido área de Procesos (PA) (véase Figura 2). El siguiente gráfico se resume el PA de Administración de Riesgos y destaca la importancia de los componentes estudiados:

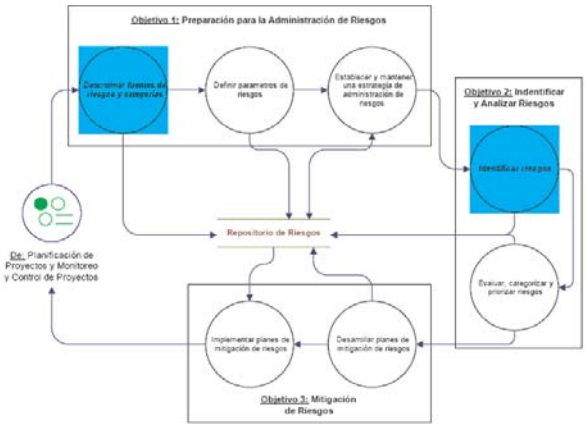

Fig. 2. PA de administración de riesgos CMMI.

La enorme importancia que los aspectos mencionados tienen en el marco de las actividades de Administración de Proyectos (y por tanto en el área de Ingeniería de Software) considerando dentro de este contexto el continuo esfuerzo realizado y la permanente y creciente necesidad presentada por las compañías de software con relación a herramientas que permitan automatizar y estandarizar sus procesos de gestión en busca de una mayor madurez organizacional.

GESTIÓN RIESGOS DEL SEI EN UN PROYECTO UNIVERSITARIO DE DESARROLLO DE SOFTWARE-SEI-CMR

Aunque los diversos enfoques de gestión del riesgo aparecieron hace más de una década, sigue siendo evidente la poca utilización de sus técnicas en los proyectos de desarrollo de software actuales. Uno de los métodos más conocido es el método del SEI, conocido como Continuous Risk Management (SEI-CRM).

En este artículo mostraremos la aplicación de este método en un proyecto universitario de desarrollo de software de grandes dimensiones. Además, nuestro trabajo muestra que conviene completar el SEI-CRM con un conjunto apropiado de riesgos organizacionales. Con nuestra investigación aplicada, esperamos contribuir al conocimiento de la gestión de los riesgos en proyectos de desarrollo del software, mediante la ampliación del método SEI-CRM con aquellos factores organizacionales de riesgo que han resultado relevantes en nuestro proyecto y en nuestra investigación previa.

El método Continuous Risk Management (SEI-CRM), desarrollado por el Software Engineering Institute (SEI), es un método en el ámbito de la ingeniería del software cuyos conceptos, procesos y herramientas permiten gestionar de manera continua los riesgos de un proyecto, proporcionando un entorno disciplinado para la toma preactiva de decisiones a lo largo de todas las fases del proyecto: análisis de los problemas en potencia (riesgos), determinación de los riesgos importantes para elaborar estrategias y planes para gestionarlos. Estos riesgos son controlados hasta que se resuelven o se convierten en problemas menores, y son tratados como tales. En la Tabla 1 podemos ver las funciones típicas de gestión del riesgo que tiene el SEI-CRM pero además este método también incluye el concepto de gestionar estas actividades como un ciclo básico, es decir, identificar, analizar, planificar, seguir, controlar y comunicar los riesgos a lo largo de todo el ciclo de vida del proyecto.

GESTIÓN DEL RIESGO EN LA FASE DE INGENIERÍA DE REQUISITOS

La ingeniería de requisitos es un área de investigación que procura atacar un punto fundamental en el proceso, que es la definición de lo que se quiere producir. Jackson afirma que la ingeniería de requisitos se ubica en el punto de encuentro entre lo informal y lo formal del desarrollo de software [Jackson, 2001].

La gestión de riesgos en el ámbito del software procura formalizar conocimiento orientado a la minimización o evitación de riesgos en proyectos de desarrollo de software, mediante la generación de principios y buenas prácticas de aplicación realista [Roppponen, 2000]. Hasta el momento se ha propuesto y utilizado diferentes enfoques de gestión del riesgo desde que Boehm [Boehm, 1988] atrajo a la comunidad de ingeniería del software hacia la gestión del riesgo. Sin embargo, es evidente que pocas organizaciones utilizan todavía de una forma explícita y istemática métodos específicos para gestionar los riesgos en sus proyectos software.

En [Pressman, 2002] se presenta la definición de riesgo dada por Robert Charette en [Charette, 1989] donde plantea que en primer lugar, el riesgo afecta a los futuros acontecimientos. En segundo lugar, el riesgo implica cambios. En tercer lugar, el riesgo implica elección, y la incertidumbre que entraña esta. Cuando se considera el riesgo en el contexto de la ingeniería de software, los tres pilares de Charette se hacen continuamente evidentes. Es indiscutible que están presentes permanentemente las características de incertidumbre (acontecimiento que caracteriza al riesgo y que puede o no ocurrir) y de pérdida (si el riesgo se convierte en una realidad ocurrirán consecuencias no deseables o pérdidas).

Están definidas las categorías de riesgos: los riesgos del proyecto, que amenazan el plan; los riesgos técnicos, que amenazan la calidad y la planificación temporal; y los riesgos del negocio, que amenazan la viabilidad del proyecto o del producto.

También son claras las estrategias frente al riesgo. Por un lado están las reactivas, cuyo método es evaluar las consecuencias del riesgo cuando este ya se ha producido (ya no es un riesgo) y actuar en consecuencia. Este tipo de estrategias acarrea consecuencias negativas, al poner el proyecto en peligro. Y por el otro las preactivas, que aplican el método de evaluación previa y sistemática de los riesgos y sus posibles consecuencias, a la par que conforman planes de contingencias para de evitar y minimizar las consecuencias. Consecuentemente, este tipo de estrategias permite lograr un menor tiempo de reacción ante la aparición de riesgos impredecibles.

Particularmente estoy de acuerdo con los partidarios de la aplicación de estrategias preactivas y coincide con [Pressman, 2002] y [Gallagher, 1999] en la necesidad de la realización de los análisis de riesgos de forma temprana, sistemática, formal y profunda.

METODOLOGIA MSF DE RIESGOS DEL MICROSOFT SOLUTIONS FRAMEWORK

Microsoft Solutions Framework (MSF) ha desarrollado un proceso para identificar y valorar ininterrumpidamente los riesgos de un proyecto, dar prioridad a estos riesgos e implementar las estrategias para tratar estos riesgos de forma proactiva a lo largo del ciclo de vida del proyecto, tal como se define en el Modelo de procesos de MSF.

Este documento presenta la información básica de la disciplina de administración de riesgos que describe los principios, conceptos, consejos, así como un proceso dividido en seis etapas para conseguir administrar con éxito los riesgos de los proyectos de TI. La información de este documento debería servir de ayuda para que un equipo de proyectos con experiencia que utiliza MSF pueda implementar un proceso proactivo de administración de riesgos para un proyecto de TI. Las personas sin experiencia en la administración de riesgos de proyectos de TI deberían ser capaces de comprender los conceptos básicos, la terminología y los principios necesarios para participar y contribuir activamente en la administración de riesgos de MSF durante el ciclo de vida de un proyecto de TI.

Dentro de MSF, la administración de riesgos es el proceso que permite identificar, analizar y solucionar los riesgos para que no se conviertan en un problema y deriven en daños o pérdidas.

Las principales características de la disciplina de administración de riesgos de MSF son las siguientes:

- Carácter global que incluye todos los elementos de un proyecto: personas, procesos y elementos de tecnología.
- Incorpora un proceso intuitivo, sistemático y reproducible para la administración de riesgos de los proyectos.
- Se aplica ininterrumpidamente durante el ciclo de vida de los proyectos.
- Su tendencia es proactiva en lugar de reactiva.
- Fomenta el aprendizaje individual y colectivo.
- Es muy flexible y puede adaptarse a una gran variedad de análisis de riesgos cuantitativos y cualitativos.

Principios básicos

La disciplina de administración de riesgos de MSF se basa en la noción de que los riesgos deben tratarse de forma proactiva, que la administración de riesgos forma parte de un proceso formal y sistemático que debe considerarse como una iniciativa positiva. Esta disciplina está basada en los principios básicos, los conceptos y la metodología más importantes de MSF. Los principios básicos de MSF pueden mejorar la administración de los riesgos de los proyectos.6 Sin embargo, los siguientes principios son especialmente importantes para La disciplina de administración de riesgos de MSF.

GESTIÓN DE RIESGOS EN RUP

Resumen de Rational Unified Process (RUP). Se describe la historia de la metodología, características principales y estructura del proceso. RUP es un producto comercial desarrollado y comercializado por Rational Software, una compañía de IBM.

El antecedente más importante se ubica en 1967 con la Metodología Ericsson (Ericsson Approach) elaborada por Ivar Jacobson, una aproximación de desarrollo basada en componentes, que introdujo el concepto de Caso de Uso. Entre los años de 1987 a 1995 Jacobson fundó la compañía Objectory AB y lanza el proceso de desarrollo Objectory (abreviación de Object Factory).

Posteriormente en 1995 Rational Software Corporation adquiere Objectory AB y entre 1995 y 1997 se desarrolla Rational Objectory Process (ROP) a partir de Objectory 3.8 y del Enfoque Rational (Rational Approach) adoptando UML como lenguaje de modelado.

Desde ese entonces y a la cabeza de Grady Booch, Ivar Jacobson y James Rumbaugh, Rational Software desarrolló e incorporó diversos elementos para expandir ROP, destacándose especialmente el flujo de trabajo conocido como modelado del negocio. En junio del 1998 se lanza Rational Unified Process.

La Gestión del proyecto es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios. La planeación de un proyecto posee dos niveles de abstracción: un plan para las fases y un plan para cada iteración.

Metodología de Riesgos en RUP

El propósito de la Planificación de Proyectos de Software es establecer planes razonables para la ejecución de ingeniería de software y para la administración de proyectos de software. Estos planes, son lo necesario para administrar el proyecto de software. Sin planes realistas, no se puede implementar un proyecto efectivo de administración.

Uno de los objetivos de RUP es asegurar que las expectativas de todas las partes son sincronizadas y consistentes. Esto es asegurado a través de evaluaciones periódicas durante el ciclo de vida del proyecto, y es documentado en el Reporte de Evaluación de Status. Este reporte es utilizado para hacer un seguimiento a información acerca de recursos (humano y financiero), mayores riesgos, progreso técnico medido a través de métricas y resultados de hitos principales.

Con RUP hacemos uso de las siguientes clases de métricas:

- Progreso (líneas de código, número de clases, puntos de función por iteración, rehacer)
- Estabilidad (tipo de rehacer, volatilidad de requerimientos o implementación)
- Adaptabilidad (costo de rehacer)
- Modularidad (extensión del impacto de rehacer)
- Calidad (velocidad de descubrimiento de defectos, densidad, profundidad e inheritancia, indicador de rehacer)
- Madurez (horas de prueba por falla)
- Perfil de desembolso de recursos (planeados versus actuales)

Los documentos RUP que contienen los planes y compromisos son:

- Casos de Negocio
- Plan de Desarrollo de Software
- Plan de Medición
- Lista de Riesgos
- Plan del Proyecto
- Plan(es) de Iteración
- Evaluación(es) de Iteración, y
- Evaluación(es) de Status

La Lista de Riesgos es un artefacto de RUP que nos provee una visión de todos los riesgos conocidos en el proyecto, y sirve como entrada para la planificación y evaluación del proyecto. Cada riesgo es descrito en función de su impacto, y un plan de contingencia será desarrollado para mitigar el riesgo en cuestión. La Lista de Riesgos es desarrollada junto con los Casos de Negocio, los cuales formarán la base para la decisión de continuar o no con el proyecto. La Lista de Riesgos es mantenida a través de todo el ciclo de vida del proyecto.

METODOLOGÍA DE RIESGOS EN ISO

EL ISO 9001 la norma establecida para medir la calidad de los productos, tiene muchos defectos cuando se aplica para un producto informático o Software.

Norma ISO/IEC 12207:2002

Es una norma de la ingeniería de software resultado del esfuerzo internacional de expertos de todo el mundo entre académicos y profesionales. Busca establecer un marco de referencia para la administración de los procesos de la ingeniería de software en el mundo. Define los procesos, actividades y tareas asociadas a los procesos del ciclo de vida del software desde la concepción hasta su retiro. Define los procesos de ingeniería de software como: "un conjunto de actividades que son realizadas por un conjunto de tareas que definen como las acciones transforman las entradas en salidas".

Evolución

- 1987. Se conforma Joint Technical Committee JTC1.
- International Organization for Standarization (ISO).
- International Electrotechnical Commission (IEC)
- 1989. Se inicia el desarrollo de la ISO/IEC 12207.
- 1995. En agosto, se publica la primera edición. Participaron en la elaboración: Alemania, Australia, Brasil, Canada, Corea, Dinamarca, España, Estados Unidos de América, Finlandia, Francia, Irlanda, Italia, Japón, Holanda, Suecia, Reino Unido y República Checa
- 2002. En mayo, se publica la enmienda 1 a modo de revisión preliminar. Se considera a: ISO/IEC 15504 (evaluación del proceso), ISO/IEC 14598 (evaluación producto), ISO/IEC 15939 (medición del software).
- Perú. Se traduce durante 2003.
- Perú. Se publica mayo 2004 NTP-ISO/IEC 12207:2004.
- Perú: En Julio 2004, el Estado Peruano oficializa su uso para estandarizar procesos y productos Software, con miras a la integración y servicios en línea (2600 entidades públicas).

CASOS DE APLICACIÓN EN MINERÍA

En la Escuela Académico Profesional de Ingeniería de Minas se ha elaborado dos proyectos en ingeniería de software para comercialización de minerales y otra para caracterización geomecánica de macizos rocosos, tanto para la minería en tajo como para la minería subterránea. En ambos proyectos, el tema de gestión de riesgos es un factor que se debe tener en cuenta y es de mucha importancia por la misma solución que brinda esta aplicación en materia de

comercialización y geomecánica que por su misma naturaleza manejan variables probabilísticas, cuyos resultados finales finalizan con un solo incremento en el factor económico, pero que este factor hace consistente a la aplicación y garantiza la robustez del futuro software minero, uno de los interfases preliminares que se viene elaborando es:

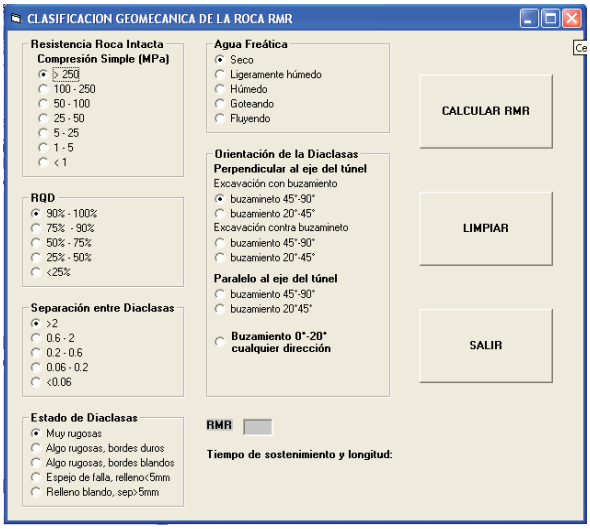

Fig. 1. Interfaz de calculo de RQD.

En esta aplicación del software minero encontramos la caracterización del macizo rocoso por medio según los índices RQD y RMR, a partir del mapeo geomecánico.

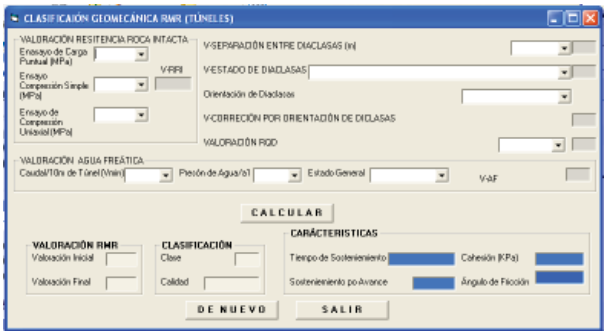

Fig. 2. Interfaz de cálculo de sostenimiento en túnel.

Finalmente en este último interfaz, se hace un analisis cuatitativo para encontrar el tiempo y tipo de sostenimiento en un macizo rocoso dependiendo de la calidad de la misma, encontrado con la aplicación inicial.

CONCLUSIONES

- La gestión de riesgos implica la toma de decisiones en la practica, el mismo que debe estar alineado al objetivo de la empresa.
- Muchas veces los riesgos aparentemente no importan, pero a mediano plazo si que importan.
- Finalmente; en la practica los riesgos que corremos en un proyecto de ingeniería de software se toman con mucha presión, cuando en la política de la empresa se debe implementar un plan de gestión de riesgos, algo que no se hace en la practica por motivos económicos.

BIBLIOGRAFÍA

1. Information Systems Audit and control Foundation (2001). "COBIT. Governance, control and audit for information and related technology". 3ª ed.
2. Jacobson, I., Booch, G., Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software.* Addison-Wesley, Madrid.
3. Pressman, R. (2001). *Ingeniería del software. Un enfoque práctico",* 5ª ed. McGraw-Hill.
4. R. M. Bernal Montañes; O. Coltell Simón (1996). *Auditoría de los sistemas de información.* Publicacions de la Universitat Politècnica de València.
5. R. Weber (1999). "Information Systems Control and Audit". Prentice Hall, Upper Saddle River, NJ.
6. Sommerville, I (2002). *Ingeniería de software.* 6.ª ed. Prentice Hall-Pearson Educación, México.
7. Australian Department of Defence. +SAFE – A Safety Extension to CMMI v1.0 (CA38809-364) (2001). Defence Materiel Organisation, Canberra, December 19.
8. International Electrotechnical Commission. Safety of machinery-Functional Safety of Safety-Related Electrical, Electronic and Programmable Electronic Control Systems (CEI IEC 62061) (2005). Geneva, Switzerland: International Electrotechnical Commission.
9. CMMI (2002) Product Development Team. SCAMPI v1.1, Standard CMMI Appraisal Method for Process Improvement, Version 1.1: Method Definition Document (CMU/SEI-2001-HB-001). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, December, 2001. http://www.sei.cmu.edu/publications/documents/06.reports/06hb002.html.