



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
DIRECCION GENERAL DE LOS ESTUDIOS DE POSTGRADO
AREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN
POSTGRADO EN GERENCIA DE PROYECTOS

TRABAJO ESPECIAL DE GRADO

**DISEÑO CONCEPTUAL DE UN SISTEMA PARA LA GESTIÓN DE RIESGOS EN
PROYECTOS DE DESARROLLO DE SOFTWARE.**

Presentado por
Marichal Barrios, Pablo Daniel.
para optar al título de
Especialista en Gerencia de Proyectos

Asesor
Abaffy Carlos

Puerto Ordaz, Mayo de 2009.

UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
DIRECCION GENERAL DE LOS ESTUDIOS DE POSTGRADO
AREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN
POSTGRADO EN GERENCIA DE PROYECTOS

TRABAJO ESPECIAL DE GRADO

**DISEÑO CONCEPTUAL DE UN SISTEMA PARA LA GESTIÓN DE RIESGOS EN
PROYECTOS DE DESARROLLO DE SOFTWARE.**

Presentado por
Marichal Barrios, Pablo Daniel.
para optar al título de
Especialista en Gerencia de Proyectos

Asesor
Abaffy Carlos

Puerto Ordaz, Mayo de 2009.

DEDICATORIA

A mi familia, gracias a su apoyo constante y leal con el cual me impulsaron para emprender este proyecto de postgrado con el objetivo de superarme profesionalmente y enriquecer mis conocimientos.

RECONOCIMIENTOS

A todos los profesores y alumnos que conocí durante el desarrollo de este postgrado, los cuales a través de sus conocimientos y vivencias, me permitieron crecer como profesional y conocer mucho más de este mundo de los proyectos en toda su extensión conceptual y humana.

ÍNDICE DE CONTENIDO

RESUMEN.....	viii
CAPÍTULO I.....	1
PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS DE LA INVESTIGACIÓN. ...	1
1. Objetivo general.....	9
2. Objetivos específicos.....	9
CAPÍTULO II.....	10
MARCO CONCEPTUAL.....	10
1. Proyecto.....	10
2. Gerencia de proyectos.....	11
3. Riesgo.....	19
4. Gerencia de los riesgos en proyectos.....	21
5. Gerencia de los riesgos en proyectos de software.....	25
CAPÍTULO III.....	42
MARCO METODOLOGICO.....	42
1. Breve introducción.....	42
2. Tipo de investigación.....	43
3. Estrategias para la Recolección, Procesamiento y Análisis de la Información.....	43
4. Factibilidad.....	54
CAPÍTULO IV.....	57
LECCIONES APRENDIDAS Y RECOMENDACIONES.....	57
REFERENCIAS BIBLIOGRÁFICAS.....	59

ÍNDICE DE FIGURAS

FIGURA 1. Competencias de demanda de un proyecto (modelo moderno).....	11
FIGURA 2. Procesos de la gerencia de proyectos.....	13
FIGURA 3. Modelo de gerencia de riesgos.....	20
FIGURA 4. Gestión de actores y ambientes.....	22
FIGURA 5. Diseño conceptual de la herramienta de software.....	32
FIGURA 6. Métodos de investigación.....	35
FIGURA 7. EDT del proyecto de software a realizar.....	42
FIGURA 8. Diagrama Gantt del proyecto.....	43

ÍNDICE DE TABLAS

TABLA 1. Competencias de demanda de un proyecto (modelo moderno).....	28
TABLA 2. Métodos de investigación vs. Métodos de desarrollo de software.....	41



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
VICERRECTORADO ACADÉMICO
DIRECCION GENERAL DE LOS ESTUDIOS DE POSTGRADO
AREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN
POSTGRADO EN GERENCIA DE PROYECTOS

“Diseño conceptual de un sistema para la gestión de riesgos en proyectos de desarrollo de software”

Autor: Pablo Daniel Marichal Barrios

Asesor: Carlos Abaffy

Año: 2009

RESUMEN DEL TEG

Se considera a la gerencia de riesgos, como un factor crítico de éxito a la hora de obtener los resultados anhelados por cualquier gerente de proyecto. A pesar de tal validez, esta área de conocimiento no es tomada muchas veces en cuenta con la actitud y la seriedad que corresponde en toda su magnitud. Esto deriva indefectiblemente en retrasos, cancelaciones, fallas operativas, sobrepaso de costos, tiempo y una gran cantidad de cosas indeseables, las cuales son conocidas por cualquier persona que ha estado involucrada en el mundo de los proyectos. La ingeniería del software no escapa a tal situación, más bien se podría decir en base a análisis estadísticos llevados por centenares de autores, es una de las áreas más susceptibles a fallar si los riesgos no son tomados en cuenta de forma sistemática y estructurada a la hora de desarrollar cualquier proyecto que derive en un sistema de información. La necesidad de contar con una herramienta automatizada que impulse la iniciativa de los gerentes de proyectos avocados al desarrollo de software a manejar los riesgos y conocer de primera mano lo que involucra la adopción de esta cultura en proyectos de este tipo, es algo vital que puede radicalmente cambiar sus paradigmas en el cómo hacer las cosas y en el cómo estos pueden actuar con respecto al tratamiento de situaciones no deseadas, las cuales no fueron previstas en el mundo ideal pensado al momento de la planificación, para así augurar un porcentaje elevado de éxito y de esta forma poder cumplir las expectativas de todos los stakeholders involucrados en el proyecto.

Palabras claves: Proyecto, Gerencia de Riesgos, Software.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS DE LA INVESTIGACIÓN.

La gerencia de riesgos no es algo nuevo en la palestra del mundo de los proyectos. Paseándose por el origen de la palabra riesgo, se encuentra que este término proviene del verbo italiano “risicare”. Fue en la época del renacimiento cuando este término nace propiamente, específicamente en el siglo XVI, la cual se consideró una época de descubrimiento para la humanidad (Wallmüller, 2005). El diccionario de la Real Academia Española 2008, lo define de esta manera: “contingencia o proximidad de un daño”. Desde este punto de vista, se puede inferir que el riesgo es algo perjudicial en cierta manera y es algo a evitar en la medida posible.

La gerencia de riesgo es un área de conocimiento dentro de los procesos de la gerencia de proyectos, la cual comprende una serie de sub-procesos que están focalizados a disminuir la probabilidad e impacto de los eventos negativos y aumentar la probabilidad de éxito en la ocurrencia de eventos positivos dentro del proyecto.

La planificación de la gerencia de riesgos en proyectos, es sin duda alguna, un proceso neurálgico dentro del marco de una fundamentada dirección de proyectos, ya que augura un gran porcentaje de éxito (bajo una correcta planificación) en la consecución del objetivo que se persigue.

Los riesgos son en sí parte fundamental de cualquier proyecto, su misma caracterización de incertidumbre y su posible consecuencia de acción positiva o negativa dentro del curso de vida de un proyecto, los hacen muy importantes a considerar, ya que su correcto manejo hacen que un proyecto sea exitoso o no en

términos de una correcta planificación de presupuesto, planificación del tiempo, fiabilidad y calidad del producto esperado (Chatterjee y Ramesh, 1999).

Los proyectos están presentes en todas las áreas del saber. Partiendo de esta premisa, una de las áreas que ha ido evolucionando constantemente, es el área de la Informática. Dentro de esta área, se encuentra el desarrollo de software, el cual no es más que la construcción de sistemas de información que permiten la ejecución de tareas a través de medios automatizados.

El software según la Institute of Electrical and Electronics Engineers, (I.E.E.E.) (2006) se define como: “la suma total de los programas de cómputo, procedimientos, reglas de documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo”. El desarrollo del software va enfocado directamente a la utilización de técnicas soportadas en las ciencias de la computación, la gerencia de proyectos, la matemática y cualquier ciencia del saber en la cual la aplicabilidad de la misma deriva en la resolución y automatización de problemas comunes.

De acuerdo a Higuera y Yacov (1996), las variables fundamentales en todo sistema de software son: la tecnología, el hardware, el software, el personal, el costo y la programación del tiempo. Todas y cada una de ellas interrelacionadas y contextualizadas entre sí, en virtud al sistema que se desea desarrollar. La complejidad de la interrelación de las variables mencionadas aumenta de manera proporcional en función de los riesgos que se presenten asociados a sí mismas.

Muchas investigaciones coinciden que el desarrollo y puesta en marcha de proyectos de software es un proceso muy complejo y que encierra muchas variables a considerar, las cuales generalmente son tomadas a la ligera o manejadas de manera incierta y algunas veces manejadas al mejor estilo de Indiana Jones; tal cual lo comenta Pressman (2001), “en las películas Indiana Jones, cuando se enfrentaba a una dificultad insuperable, siempre decía, << ¡no te

preocupes, pensaré en algo! >>” (p. 98), lo cual evidencia que el desconocimiento de una situación imprevista posee una gran probabilidad de ocurrencia generalmente en términos indeseables para el proyecto, y en la realidad no todo termina en como en las películas, con un final feliz.

Sumado a esto, tal como lo comenta Palacios (2005), “en la cultura latina la identificación de riesgos es un proceso muy débil en la metodología, dado que se tiende a ser muy optimista en las estimaciones, con suposiciones poco fundamentadas en datos históricos y estadísticos” (p. 421), lo cual no hace más que añadir una variable mas al conjunto de características que identifican a los gerentes de proyecto latinos avocados al desarrollo de software.

El desarrollo del software encierra en sí mismo una alta complejidad, ya que no es algo que se fabrica, es algo que no se estropea pero se deteriora en el tiempo, es algo también que se construye a la medida de las necesidades de un usuario o grupos de usuarios con necesidades similares que son manejadas bajo su contexto de acción (Pressman, 2001).

Características de esta naturaleza evidencian la necesidad y el origen de la ingeniería de software, la cual se encarga de proporcionar un marco de trabajo claro, efectivo y preciso para construir software de calidad. Esta rama de la ciencia de la computación hace gran hincapié en la necesidad de contar con una gerencia de riesgos a todo nivel, durante el desarrollo de un proyecto de software de cualquier envergadura.

Taylor (2004), explica que los proyectos tecnológicos, caracterizados por el uso de tecnologías propiamente variadas y tan cambiantes en el tiempo, ameritan definir el riesgo en relación a un contexto propio de acción, lo cual hace caracterizar al riesgo bajo tres componentes fundamentales: el evento (¿qué puede pasarle al proyecto?, ¿es bueno o malo?), la probabilidad de ocurrencia del

mismo (¿cuáles son las posibilidades de que el evento ocurra?) y el impacto sobre el proyecto (¿cuál es el efecto?).

De esta manera el autor señala que los riesgos deben ser clasificados bajo dos tipos: los riesgos del negocio (basados en la oportunidad de la ganancia o pérdida con el proyecto) y los riesgos puros o asegurables (son los que hay que evitar a toda costa, tales como los desastres naturales). La clave según el autor, es reconocer los eventos potenciales de riesgo y el cómo saber direccionar o controlar el efecto de una manera neutral o positiva dentro del proyecto.

Tal escenario de complejidad deriva indefectiblemente en una incertidumbre de finalización efectiva del proyecto bajo aspectos de estar bajo el presupuesto destinado y el tiempo prometido (Masticola, 2007). En dicho estudio el autor hace referencia al “Standish Group’s CHAOS Report” del año 2004, el cual cuenta para ese momento, con una base de información de 50.000 proyectos de software completados para la industria gubernamental y comercial, de los cuales solo el 29% fue completado con éxito, mientras que el 53% fue entregado sobrepasando tiempo o presupuesto y un fatídico 18% el cual no entregó resultados utilizables bajo ningún concepto.

No es por mera casualidad que Pressman (2001), cita que: “muchos observadores de la industria (incluyéndose este autor) han caracterizado los problemas asociados con el desarrollo del software como una <<crisis>>” (p. 98). Tales afirmaciones han sensibilizado a muchos gerentes de proyectos de la industria del software, los cuales se han visto en la necesidad de implementar algún mecanismo que les permita controlar de cierta manera los eventos de incertidumbre con carácter de desviación negativo en lo que es la realización de todas las fases de un proyecto de esta naturaleza.

Suri y Wadhwa (2007) señalan que: “el desarrollo de software se encuentra lleno de riesgos. Desde momentos tempranos como el inicio de la propuesta, el desarrollo, el testeo de la aplicación, existen una multitud de factores que introducen el riesgo. Los proyectos grandes de desarrollo de software tienen la más alta probabilidad de ser cancelados o congelados en algún momento por la acción de estos factores.”

La gerencia de riesgos es una disciplina que ha venido tomando forma en el tiempo y cada vez se encuentra más madura en términos de implementación de un lenguaje propio, contando con técnicas e instrumentos ampliamente aceptados, que permiten de una forma proactiva adquirir un enfoque sobre el manejo del factor de incertidumbre, de manera clara y concisa, enfocándose en la obtención de los resultados esperados (Hilson, 1998). Reforzando la idea anterior Lister (2003), lo plantea de la siguiente manera: “la gerencia de riesgos es la gerencia de proyectos de adultos”, y cabe destacar que dicha afirmación ha pasado a ser parte de la jerga de la gerencia de proyectos.

Partiendo del hecho indiscutible de que los proyectos de software manejan un número considerable de factores de incertidumbre a lo largo de su ciclo de vida, muchos investigadores, organizaciones gubernamentales y comerciales han desarrollado mecanismos de implementación, paradigmas, marcos de trabajo enfocados a la gerencia de riesgos que les ayude a superar los nuevos retos que emprenden en lo que es el desarrollo de un software, el cual indefectiblemente siempre está plagado de elementos de este tipo.

Tomando modelos de otras áreas del saber, tal como es el caso de las finanzas, en donde la gerencia o el manejo del riesgo son algo ineludible y el grado de sofisticación de esta disciplina se ha ido incrementando durante su misma evolución en el tiempo, ciertas técnicas e instrumentos han sido copiados e implementados para su reutilización dentro de lo que son muchos estudios de

gerencia de riesgo aplicados a proyectos de software, tal es el caso de estimación de costos por fallas en proyectos de software (Masticola, 2007), el cual fue desarrollado utilizando información fácilmente recogida de estimaciones de costos de proyectos de software que habían fallado, la cual se usó para estimar la tasa de calidad que debería tener el proyecto a fin de evitar el fracaso reflejado económicamente.

En varias investigaciones (Pressman 2001; Higuera y Haimés, 1996; Wallmüller, 2005; Miler y Górski, 2001; IEEE Standar 1540, 2001), por citar algunos autores, plantean un modelo de gerencia del riesgo determinado por una serie de estrategias, representadas en procesos que permiten la implementación de una gerencia efectiva sobre los riesgos de un proyecto, enfocados en este caso a los de software. Entre las más importantes estrategias o sub-procesos se tienen: la identificación, el análisis, la priorización, la evaluación, seguimiento, control y monitoreo del riesgo. Los autores determinaron conceptualmente que la gerencia de riesgos está determinada bajo un esquema similar de iteración continua de sub-procesos dentro del ciclo de vida del proyecto. Estos modelos responden y se familiarizan entre sí en todos los casos, unos más complejos que otros, pero encerrando los aspectos claves de la gerencia del riesgo.

Cada uno de estos procesos englobados en un modelo de gerencia de riesgos, responden a actividades a ejecutar de manera cíclica durante la vida del proyecto. Bajo ese marco contextual de trabajo, se pueden utilizar herramientas y técnicas en búsqueda de conseguir los resultados esperados, que permiten la canalización de los riesgos de una manera idónea en la mayoría de los casos, asegurando una gran probabilidad de éxito al momento de su implementación.

Cabe destacar que algunos modelos de gerencia de riesgos anexan como sub-proceso final la documentación y almacenamiento del riesgo, identificado al inicio del proceso, a fines de ir creando una gestión de conocimientos de gerencia de

riesgos para futuros proyectos de una misma índole (Taylor, 2004), derivando incluso en sub-procesos de “aprendizaje de riesgos” tal como es señalado por el MSF Risk Management Discipline v.1.1 en su sub-proceso 6 del modelo de gerencia de riesgos desarrollado por Microsoft (2002).

La necesidad de estructurar un plan real de gerencia del riesgo enfocado al proyecto es algo tajante y a grandes rasgos un concepto a buscar en todo proyecto de software. La implementación de métodos prácticos para la identificación y análisis de riesgos incluyendo una taxonomía usada para organizarlos dentro del proyecto, corresponde el crear un marco de trabajo a seguir, con esas características señaladas.

La implementación derivada de estrategias de riesgo, la documentación, la comunicación efectiva de los riesgos del proyecto a cada uno de los stakeholders, la identificación de roles claves y responsabilidades en la gerencia del riesgo del proyecto, son parte de estas actividades complementarias que derivan en una correcta manera de ejercer la gerencia de riesgos en términos de desarrollo de software y de cualquier clase de proyectos, los cuales deben de seguir un patrón de comportamiento contemplando cada uno de los sub-procesos mencionados hasta el momento.

Partiendo de tales premisas y su asociación directa a lo que respecta la necesidad de la gerencia de riesgos aplicada al desarrollo de software, surge la necesidad de contar con un instrumento que permita brindar tanto al gerente de proyecto, como también a los stakeholders del mismo, la capacidad de gestionar los factores de incertidumbre, bajo un marco de trabajo, conceptualizado y colaborativo de gerencia de riesgos, siguiendo modelos de software aplicados por Karolak (1997) en su sistema SERIM “Software Engineering Risk Management”, conjuntamente con el sistema desarrollado por Miller y Górski (2001), el cual se fundamenta en el uso de sistemas computacionales diseñados para ser utilizados

a través de interconexión de redes, tal como es el caso de la Internet, a fines de expresar el aspecto colaborativo, algo fundamental en la gerencia de riesgos, en la cual todos los stakeholders del proyecto puedan participar intensamente en cada uno de los distintos sub-procesos de la gerencia de riesgos e ir aportando sus ideas al respecto, ayudando de esta manera a la mitigación e identificación de riesgos de manera temprana.

Esto se debe lograr al utilizar e implementar las mejores prácticas utilizadas por los distintos autores ya citados, a fines de instruir e inducir a utilizar los beneficios de esta área de conocimientos de la gerencia de proyectos, en pro de la consecución de todos los objetivos planteados por el proyecto de software que se pudiera emprender en un momento determinado, manejando el factor incertidumbre de una manera clara conceptualmente y enmarcada en los lineamientos de los objetivos específicos de este planteamiento.

La utilización de un instrumento desarrollado por López (2007) el cual va dirigido a la evaluación de la gestión de riesgos en proyectos, será conceptualizado dentro de este marco de trabajo a desarrollar en la estructura de este sistema de información, a fin de reforzar los principales conceptos expuestos por el Project Management Institute (PMI) (2004), en cuanto a la gerencia de riesgos en proyectos se refiere.

De tal forma está explícita la formulación de la pregunta que antecede a la elaboración de este proyecto en cuestión, la cual es la siguiente:

¿Cuáles serían las características de un software para la gestión de riesgos en proyectos de desarrollo de software?

Ante tal necesidad surge de plantear el objetivo general del proyecto y sus objetivos específicos:

1. Objetivo general

Diseñar conceptualmente un sistema de información que permita la gestión de la gerencia de riesgos en proyectos de desarrollo de software, basado en las mejores prácticas de ésta área de conocimientos de la gerencia de proyectos.

2. Objetivos específicos

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

- Planificar la gestión y el alcance del proyecto.
- Implementar un marco definido de trabajo conceptualizado bajo las metodologías más destacadas de la gerencia de riesgos en proyectos de software para la creación de la herramienta.
- Crear un diseño detallado que sustente toda la información y funcionamiento del proyecto enmarcado en las características establecidas.
- Utilizar los patrones de la arquitectura de software referenciados a la metodología XP de programación extrema para un desarrollado acelerado de la herramienta en concordancia de los atributos planteados.

CAPÍTULO II

MARCO CONCEPTUAL

El planteamiento del problema deja en claro la importancia de la gerencia de riesgos en el desarrollo de software. Esta es un área de conocimiento que actualmente está muy en boga dentro del mundo proyectos y está ampliamente explotada y profundizada a la hora de implementar una adecuada gerencia de proyectos. Esto refleja el hecho meritorio de contar con un marco de conceptos que dejen claro al lector desde lo más simple hasta lo más denso, a manera de explicar a grandes rasgos el desarrollo de este trabajo de investigación.

1. Proyecto

Desde un punto de vista primigenio, el concepto a referir por excelencia antes de adentrarse en lo específico que caracteriza a esta investigación es el término de proyecto. El PMI (2004), lo define de la siguiente manera: “esfuerzo temporal para crear un producto, servicio o resultado único” (2000, p. 4).

López (2007), lo define de la siguiente forma: “proceso único consistente en un conjunto de actividades coordinadas y controladas con fechas de inicio y de finalización, llevadas a cabo para lograr un objetivo conforme a requisitos específicos, incluyendo las limitaciones de tiempo, costo y recursos”.

A partir de estos conceptos encontramos muchas variantes del mismo término, pero en realidad todas apuntan y se centran esencialmente en lo expuesto.

2. Gerencia de proyectos

La necesidad de conocer la naturaleza y el funcionamiento a nivel de procesos de lo que implica en toda la extensión del término proyecto, ha hecho marcar la necesidad de crear la manera de que este tipo de actividades orientadas a un único fin y con tiempo y recursos determinados, puedan ser controladas y administradas en un marco de trabajo probado a través de mejores prácticas y consideraciones a lo largo de los años, lo cual ha devenido en lo que se conoce actualmente como la gerencia de proyectos.

Es de conocimiento general que normalmente en cualquier aula de clases o libros de texto enfocados a la gerencia de proyectos, se haga alusión de que el inicio de esta actividad sea remontada al período posterior de la segunda guerra mundial, esto dado al hecho de que el desarrollo militar requirió de alguna manera la implementación de tareas complejas y organizadas bajo parámetros de tiempo y costos definidos.

Herramientas como los diagramas de barras o Gantt, PERT (Program Evaluation and Review), CPM (Critical Path Method), fueron producto de las investigaciones que dieron origen a la utilización de herramientas que sustentaran de manera científica y matemática el surgimiento o puesta en marcha de esta actividad a toda escala y a nivel mundial.

Taylor (2004), difiere un poco en lo que al surgimiento de la gestión de proyectos se refiere, ya que comenta que la gerencia de proyectos es una tarea que ha existido en la humanidad a lo largo de miles de años. Su aseveración radica en la exposición de ejemplos prácticos, tales como las grandes pirámides en Egipto, los templos y palacios creados por Salomón y los magníficos programas de construcción tanto de los griegos como de los romanos. En gran parte de su exposición se evidencia la verdad que acompaña el hecho de que

estas culturas contaron con instrumentos que en cierta manera le permitieran coordinar tales magistrales obras, que por solo citar una de ellas, la gran pirámide de Keops, la cual consta de 2.300.000 de bloques de piedra con un peso de 2 toneladas cada uno de ellos.

Sin lugar a dudas Taylor (2004), deja claro que la gerencia de proyectos no nació de la nada y mucho menos de un momento para otro. Es obvio que también el advenimiento de la era industrial fue el empujón más drástico y significativo para darle forma como disciplina y área de conocimiento de las ciencias gerenciales del momento. Esto ocurre principalmente con la evolución tecnológica de los proyectos en sí mismos y la necesidad de contar con herramientas y técnicas, tales como las que se mencionaron anteriormente aparecidas después de la segunda guerra mundial, que permitiesen llevar a cabalidad una correcta gerencia de un proyecto en toda su extensión.

La gerencia de proyectos, viéndolo desde un punto de vista conceptual, puede definirse como el conjunto de técnicas y herramientas, ejecutadas por personas, para describir, organizar y monitorear el trabajo de las actividades de un proyecto. La necesidad imperante de contar con un marco racionalizado de trabajo para el emprendimiento de un proyecto por parte de muchas empresas y organismos estatales a nivel mundial dieron pie a su evolución acelerada.

Viendo los resultados que un marco de trabajo de estas características brindaba a cada uno de los proyectos que eran sometidos, fueron buscando el perfeccionamiento e inclusión de nuevas técnicas que permitieran manejar todo el contexto que encierra la gerencia de proyectos. Esto queda evidenciado por lo expresado por Graham y Englund (1999) los cuales lo comentan de esta manera:

“... las metodologías de proyectos y el entrenamiento en gerencia de proyectos permiten, entre otros aspectos, que los directivos establezcan patrones de

medición de logro, enfoquen sus actividades a los clientes, cuantifiquen el valor de la organización, optimicen el uso de recursos, incorporen principios de calidad, implementen nuevos planes estratégicos, mejoren su desempeño y respuesta al mercado, innoven nuevos productos y desarrollos, mejoren el nivel de conocimiento y capacitación de sus empleados y mejoren sus niveles administrativos.”

El uso de este marco de trabajo dentro de los lineamientos dictaminados por las mejores prácticas, en conjunto con la experticia propia del personal humano y los recursos necesarios para la realización del proyecto, en cierta manera otorgan una seguridad a cualquier sponsor de un proyecto en lo que refiere a la culminación del mismo (respetando costo, calidad y tiempo), esto siempre bajo el equilibrio de las exigencias demandadas.

Dichas exigencias están enmarcadas en el nuevo enfoque que se ha originado en la gerencia de proyectos, las cuales se contextualizan entre las relaciones de las competencias naturales de un proyecto, estas no son más que las que se muestran en la figura siguiente, extraída de Horine (2005):

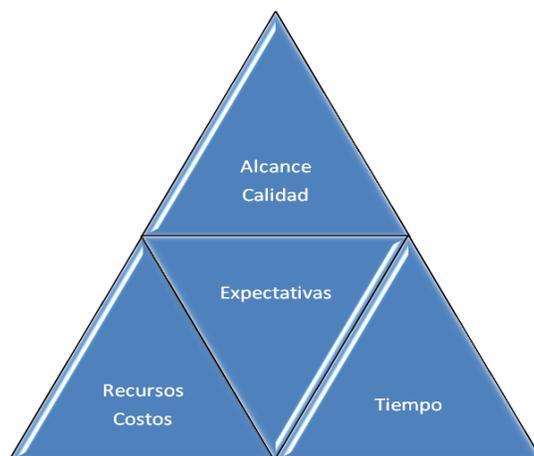


Figura 1. Competencias de demanda de un proyecto (modelo moderno). (Horine, 2005)

Dichas competencias se originan en virtud a lo que se exige de un proyecto a término de su cierre propiamente. Esto significa que el proyecto en base a un alcance determinado y una calidad esperada sustentado a expectativas previas, debe realizarse a un costo y tiempo específico, utilizando los recursos disponibles. De esta manera se triangula la idea en base a las expectativas originadas a fin de que estas sean satisfechas una vez culminado el proyecto.

La gerencia de proyectos según Heerkens (2001), se resume en dos aspectos fundamentales a tomar en cuenta: el arte y la ciencia. El “arte”, el autor lo refiere al hecho de lidiar con la gente tratando de que estas logren el cumplir la realización de las metas, lo cual es algo a tomar en cuenta realmente. Es un arte también porque se debe poseer un profundo conocimiento del comportamiento humano para saber dirigirlo al aplicar las correctas habilidades interpersonales en el contexto del desarrollo de cualquier proyecto en un momento dado. Mientras que lo que refiere a “ciencia”, se enfoca sustancialmente al conocimiento, comprensión y a la habilidad para la aplicación de los procesos de la gestión de proyectos de manera prescrita. Este proceso tiene por guía que los directores de proyectos y equipos de proyecto implementen de manera eficaz el desempeño de las principales actividades del proceso, basándose en tareas tales como la determinación de la verdadera necesidad, definiendo el objetivo del proyecto, creando un cronograma de ejecución y manteniendo un control a través de la vida del proyecto.

La gerencia de proyectos es una disciplina que ha pasado a formar parte esencial de la administración moderna de toda empresa, destacándose a todo nivel, desde largas corporaciones hasta empresas de carácter familiar inclusive.

La evolución más notable de la gerencia de proyectos se ve reflejada en los años noventa, sin duda alguna una década que ha marcado pauta en el tiempo en

cuanto a al desarrollo de la humanidad, lo cual lo expone de manera bien clara Solarte (2003), de la siguiente forma:

“la gestión de proyectos en las organizaciones, a partir de la década de los noventa, se convierte en una importante herramienta para la consecución de objetivos mediante la conformación de estructuras flexibles, livianas y de carácter temporal, lo cual contribuye a la disminución de sus costos y a la consecución de los resultados de manera eficaz”.

Adentrándose cada vez más en esta disciplina y constatando su alto nivel de especialización en virtud de la naturaleza intrínseca de los proyectos, se encuentra la subdivisión de los procesos en términos de dirección de proyectos, expuestos por el PMI (2004), la cual lo representa en cinco grupos fundamentales:

- **Grupos de procesos de iniciación**, estos definen y autorizan el proyecto o una fase del mismo.
- **Grupos de procesos de planificación**, se dirigen a definir y refinar los objetivos y planifica el curso de acción requerido para lograr los objetivos y el alcance pretendido del proyecto.
- **Grupos de procesos de ejecución**, enfocados en integrar a personas y otros recursos para llevar a cabo el plan de gestión del proyecto para el proyecto en sí mismo.
- **Grupos de procesos de seguimiento y control**, destinados a medir y supervisar regularmente el avance, a fin de identificar las variaciones respecto del plan de gestión del proyecto, de tal forma que se tomen medidas correctivas cuando sea necesario para cumplir con los objetivos del proyecto.

- **Grupos de procesos de cierre**, están encargados de formalizar la aceptación del producto, servicio o resultado y termina ordenadamente el proyecto o una fase del mismo.

De forma clara puede ser visto como interactúan dichos grupos de procesos a lo largo del ciclo de vida del proyecto a través de la siguiente figura, extraída del PMI (2004):

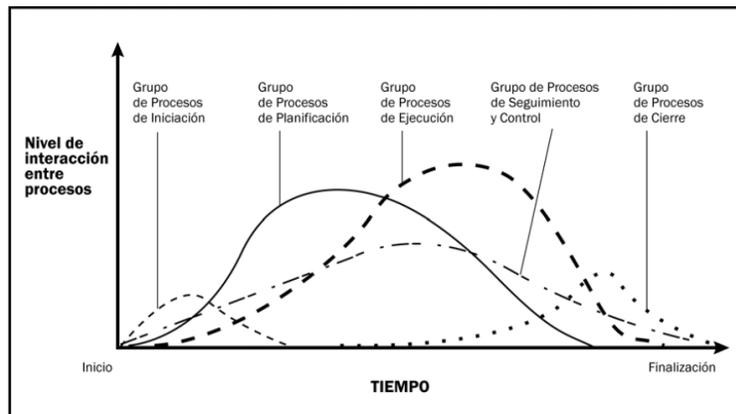


Figura 2. Procesos de la gerencia de proyectos (PMI, 2004)

Todo proyecto está estructurado por una fase de inicio, una fase intermedia y una fase de culminación, cada una corresponde o simboliza la conclusión y aprobación de uno o más productos entregables dentro del proyecto. De esta manera se puede observar que cada proceso, independiente uno de otro, puede estar sobre una fase determinada en sí mismo ante el proyecto y a su vez otro proceso puede estar contemplado y en ejecución de manera paralela e interdependiente en un instante de tiempo determinado a través de la fase en la cual se encuentre.

La gerencia de proyectos está conformada a través de nueve áreas de conocimiento, las cuales representan procesos de entrada, herramientas y salidas destinadas a disciplinas específicas dentro del proyecto pero siempre

cohesionadas entre sí, con la finalidad de brindar el marco metodológico y estructural que permita el dimensionamiento adecuado a las necesidades del proyecto en términos de magnitud y alcance para el logro de los objetivos planteados.

Estas áreas de conocimiento están delimitadas de acuerdo a campos de acción dentro del proyecto. Cada área se focaliza, se especializa en su terreno de acción y se desagrega a sí misma, para aislarse dentro de lo que es el entorno del proyecto y así de esta manera pueda ser atacada en términos de planificación, ejecución y control a través del uso de herramientas específicas con la finalidad de permitir la consecución de entregables que den paso al engranaje que representa la gerencia de proyectos.

Las áreas de conocimiento se clasifican de esta manera según el PMI (2004):

- ***Gestión de la integración del proyecto:*** Se enfoca a los procesos y actividades necesarios para identificar, definir, combinar, unificar y coordinar los distintos procesos y actividades de dirección de proyectos dentro de los grupos de procesos de dirección de proyectos.
- ***Gestión del alcance del proyecto:*** Se avoca a asegurar que el proyecto incluya todo el trabajo requerido, y solo el trabajo requerido, para completar el proyecto satisfactoriamente. Está relacionada con la definición y control de lo que esta y no está incluido en el proyecto.
- ***Gestión del tiempo del proyecto:*** Se encarga de incluir todos los procesos necesarios para lograr la conclusión del proyecto a tiempo.
- ***Gestión de los costes del proyecto:*** Esta remitido a los procesos involucrados en la planificación, estimación, preparación del

presupuesto y control de costes de forma que el proyecto se pueda contemplar dentro del presupuesto aprobado.

- ***Gestión de la calidad del proyecto:*** Son aquellas actividades de la organización ejecutante que determinan las políticas, los objetivos y las responsabilidades relativos a la calidad de modo que el proyecto satisfaga las necesidades por las cuales se emprendió.
- ***Gestión de los recursos humanos del proyecto:*** Enfocada a incluir los procesos que organizan y dirigen el equipo del proyecto. El equipo de proyecto está compuesto por las personas a quienes se les han asignado roles y responsabilidades para concluir el proyecto.
- ***Gestión de las comunicaciones del proyecto:*** Es el área que lleva todos aquellos procesos necesarios para asegurar la generación, recogida, distribución, almacenamiento, recuperación y destino final de la información del proyecto en tiempo y forma.
- ***Gestión de los riesgos del proyecto:*** Está relacionada a todos los procesos de planificación de la gestión de riesgos, la identificación y el análisis de riesgos, las respuestas a los riesgos, y el seguimiento y control de los riesgos en un proyecto, los cuales funcionan en una sincronía conjunta y se actualizan durante la vida del proyecto.
- ***Gestión de las adquisiciones del proyecto:*** Se referencia a todos los procesos para comprar o adquirir los productos, servicios o resultados fuera del equipo del proyecto para realizar el trabajo.

Teniendo claro la caracterización de cada una de las áreas de conocimiento que encierra la gerencia de proyectos, el lector puede tener un conocimiento

general del alcance de la misma, en términos de productividad, acción y ejecución en el ámbito organizacional e industrial en que se aplique, siempre mirando a la necesidad de llevar a término exitoso un proyecto de cualquier naturaleza.

De esta forma se puede observar que la gestión segmentada por áreas de conocimientos permite al gerente del proyecto estructurar un marco de trabajo lo suficientemente funcional y racional en cuanto al manejo de procesos, de tal forma que no se le haga imposible lidiar con cada uno de los elementos durante el ciclo de vida del proyecto.

A fin de reforzar el enfoque del presente trabajo de investigación, es necesario profundizar en todo lo que respecta al área de la gestión de riesgos en los proyectos. Un área que sencillamente es fundamental y debe tomarse en cuenta en el proyecto que sea, por más pequeño en términos de recursos y tiempo que este contemple.

El simple hecho de no hacerlo, puede desembocar en el origen de desviaciones impositivas en cuanto al correcto curso que debería seguir la programación del proyecto a raíz de la aparición de eventos que lo afecten y estén fuera de lo que se contempló como “curso lógico” del mismo, que generalmente en la vida real siempre pasa que no se cumpla el sueño utópico de que todo salga tal cual se planeo.

3. Riesgo

Dentro de las nueve áreas de conocimiento, se encuentra una en particular la cual es motivo de estudio a raíz de la investigación propuesta, esta no es más que la gestión de los riesgos del proyecto. Partiendo de la definición que se expuso en medio de la clasificación de las áreas de conocimientos, es necesario aportar un poco mas de detalle en cuanto a lo que abarca esta área de conocimiento en

especial dentro de la gerencia de proyectos, sus orígenes, como se estructura internamente, la definición, la creación de un marco de trabajo definido para su aplicación, el por qué del estudio de los riesgos y su aplicación al campo de los proyectos a cualquier nivel para finalmente culminar con la especialización de esta área en todo lo relacionado a proyectos de software.

Para tener un concepto más detallado de lo que es el riesgo y cómo la naturaleza del mismo incide en el ciclo de vida de un proyecto haciendo que este pueda llegar al fracaso en cualquiera de sus tres ejes principales (costo, tiempo y alcance) siempre bajo unas expectativas previstas, se puede citar la definición de Palacios (2005), la cual lo resume con gran capacidad de síntesis:

“el riesgo es una medida de la falta de certidumbre basada en la indisponibilidad de información adecuada”. (p.316).

Por otro lado y reforzando la definición anterior, López (2007) realiza una cita del PMI (2004), en la cual define el riesgo en un proyecto de la siguiente manera:

“El riesgo en un proyecto es un evento o una condición, que si ocurre, tiene un efecto positivo o negativo sobre los objetivos del mismo. Un riesgo tiene una causa y, si ocurre, una consecuencia. Los riesgos del proyecto incluyen tanto las amenazas a sus objetivos como las oportunidades de mejora de dichos objetivos. Esto tiene origen en la incertidumbre que está presente en los proyectos.” (p.127-128).

Por otro lado y viendo el riesgo desde un punto de vista tecnológico, nos encontramos con la definición del SEI (*Software Engineering Institute*) el cual lo define al riesgo como “la posibilidad de sufrir una pérdida”, SEI (2004).

Los riesgos son algo de lo que no se puede escapar. Al observar el factor riesgo, es lógicamente comprensible establecer a priori, que un proyecto sin

riesgos es algo muy difícil de encontrar, ya que el riesgo forma parte de la naturaleza propia del proyecto.

El carácter temporal del proyecto incide de tal forma en cómo la incertidumbre pudiera atacarlo en un instante de tiempo dado, evidenciado a través de la toma de decisiones (algo vital dentro del ciclo de vida de un proyecto).

En base a lo expresado a la acción de tomar decisiones en función del riesgo presentado, el riesgo puede ser atacado de dos formas:

- **Reactivamente:** como por ejemplo cuando se evalúa las consecuencias del riesgo, en este caso ya este dejó de serlo y por lo tanto se actúa a consecuencia de su acción. Este tipo de respuesta pone en riesgo el proyecto y no es la manera más aconsejable de actuar.
- **Proactivamente:** cuando se hace mediante la evaluación previa y sistemática de los riesgos, se evalúan las consecuencias, se genera un plan de evitación y se minimizan las consecuencias posibles y se crea un plan de contingencia. De esta manera se evade el riesgo y se obtiene un menor tiempo de reacción en caso de ocurrir.

4. Gerencia de los riesgos en proyectos

Según López (2007), la gerencia de los riesgos de un proyecto tiene como objetivo la conservación de los activos y el poder de generación de beneficios mediante la minimización, a largo plazo, del efecto financiero de las pérdidas accidentales. Si se analiza esta definición el autor hace prevalecer la idea de que la gerencia de riesgos es la función empresarial que permite evitar desagües financieros al momento de emprender proyectos dentro de la organización y

sencillamente eso es lo que busca cualquier gerente dentro de una organización competitiva.

Esta conceptualización expuesta sobre la gestión de los riesgos, tiene su origen primario al hecho fundamental de que la gerencia de riesgos antes de ser una tarea incorporada a la gerencia de proyectos y establecerse como un marco de trabajo sólido y definido y convertirse en un área de conocimiento propiamente, ya esta hace más de un siglo estaba presente en la industria financiera de todo mundo y venía siendo utilizada ampliamente, valga la aclaratoria aplicándose a bienes raíces, fondos económicos y actividades económicas diversas. Al ser una disciplina madura en dichos términos, al acoplarse al mundo de los proyectos, se trajo consigo misma muchas de las técnicas y herramientas hasta inclusive parte del lenguaje técnico, que han servido para propósitos financieros desde sus orígenes.

Desde un punto de vista tecnológico la manera más acertada en cómo se define la gerencia de riesgos en proyectos de software es a través de la exposición del término desarrollado por la SEI (2004), el cual refiere como: “el proceso formal en el que los factores de riesgos son sistemáticamente identificados, evaluados y mitigados”.

La gerencia de los riesgos se focaliza en determinar lo que es el tratamiento del riesgo, lo cual no es más que la serie de acciones a ejecutar una vez se pueda presentar el riesgo dentro del proyecto. Existen varios enfoques para el tratamiento del riesgo, entre estos se pueden mencionar el expuesto por la Universidad de Salamanca “Curso de gerencia de riesgos y seguros en la empresa” (1999):

- Eliminación (posible).
- Reducción.

- Retención o asunción.
- Cesión o transferencia.

También la norma ISO 10006-2003 establece un apartado para el tratamiento del riesgo de la siguiente manera:

- Eliminar.
- Mitigar.
- Transferir.
- Compartir.
- Aceptar.

Sin ser muy analítico se evidencia la similitud entre las propuestas y se infiere que al implementarlas de manera metódica y organizada, dependiendo de la situación de riesgo obviamente, permitiría al gerente del proyecto contar con las herramientas adecuadas para resolver los factores de riesgo que se pudieran estar presentando a lo largo de la vida del proyecto.

Una vez clara la manera en cómo se deben tratar los riesgos, lo más idóneo es contar con procesos que permitan atacar dichos riesgos de una manera metódica y eficaz, para lo cual no hay más que referenciar al PMI (2004), siguiendo con los conocimientos que expone en materia de gerencia de los riesgos, la cual queda subdividida en seis procesos:

- ***Planificación de la gerencia de riesgos:*** Proceso encargado de decidir cómo enfocar, planificar y ejecutar las actividades de gestión de riesgos para un proyecto.

- **Identificación de los riesgos:** Se refiere a cómo determinar qué riesgos pueden afectar al proyecto y documentar sus características.
- **Análisis cualitativo de los riesgos:** Enfocado en analizar numéricamente el efecto de los riesgos identificados en los objetivos generales del proyecto.
- **Análisis cuantitativo de los riesgos:** Se centra en desarrollar opciones y acciones para mejorar las oportunidades y deducir las amenazas a los objetivos del proyecto.
- **Planificación de la respuesta a los riesgos:** Encargada de desarrollar opciones y acciones para mejorar las oportunidades y reducir las amenazas a los objetivos del proyecto.
- **Seguimiento y control de los riesgos:** Su finalidad es realizar el seguimiento de los riesgos identificados, supervisar los riesgos residuales, identificar nuevos riesgos, ejecutar planes de respuesta a los riesgos y evaluar su efectividad a lo largo del ciclo de vida del proyecto.

De esta manera se logra representar el modelo de gerencia de los riesgos de una forma cíclica, en la cual intervienen los distintos procesos, uno a uno, a manera de formar una estructura cónsona que delimite un marco de trabajo en pos del tratamiento de los riesgos de manera clara y eficiente, augurando un éxito en la consecución del objetivo planteado por el proyecto.

A continuación se presenta a manera grafica, la gerencia de los riesgos siguiendo el modelo propuesto por el PMI (2004):

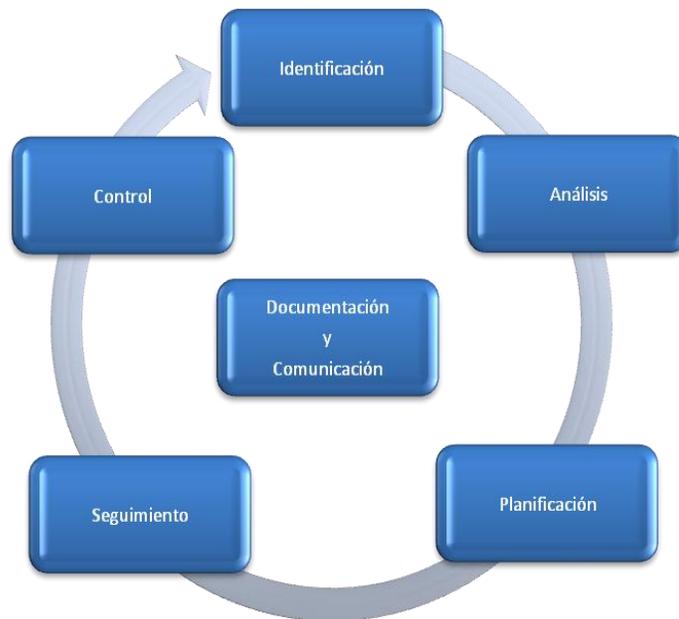


Figura 3. Modelo de gerencia de riesgos (PMI, 2004)

La necesidad de contar con un marco de trabajo que permita gestionar todos estos distintos procesos de manera sincronizada y efectiva es parte obligatoria de toda gerencia de proyectos en la que se desee lidiar con el factor incertidumbre dentro del tiempo estipulado para la finalización del mismo y los recursos que por lo general son limitados. Esto ha llevado en cierta manera a que esta área de conocimientos tome un gran auge dentro de la gerencia de proyectos y sea ampliamente estudiada y caracterizada dependiendo del tipo de proyecto en la cual se aplique.

5. Gerencia de los riesgos en proyectos de software

Al hablar de la gerencia de riesgos en proyectos de software, hay que remontarse al inicio de los años ochenta tal como lo citan Evans y Galorath (2006), cuando Barry Boehm, un líder experto en el desarrollo de software, desarrolló y publicó un modelo de manejo de riesgos que capitalizaba la naturaleza

evolucionista del proceso de desarrollo de software en un modelo de vida cíclico que permitía la evolución y crecimiento de los requerimientos mientras se minimizaban los riesgos de los cambios incontrolables de los cuales la industria del software se encuentra plagada desde sus inicios.

Para aplicar la gerencia correcta de los riesgos de un proyecto de software, es necesario e imprescindible contar con un marco de trabajo que delimite y permita conceptualizar el alcance de cada uno de los factores que aglutina dicha área de conocimiento.

En base al arqueado bibliográfico desarrollado para la elaboración de esta investigación, se encontraron muchos marcos de trabajo, que convergían en cierto punto a lo mismo y repetían en cierta manera muchos aspectos unos de otros, salvo el presentado por Lyytinen, Mathiassen y Ropponen (1996), el cual se enfoca principalmente en que el marco de trabajo depende de cómo los actores, la estructura, la tecnología en el ambiente gerencial se encuentran ensambladas en un todo.

Este marco de trabajo se centra en el rendimiento obtenido, producto del ensamble citado, el cual se define por la capacidad de procesamiento de la información disponible en los distintos aspectos que lo conforman, para saber de esta manera los incidentes críticos representados por riesgos, con el fin de sugerir alternativas para calcular las consecuencias y así tratar los riesgos naturales del software.

Entre los tres ambientes dentro del marco del proyecto se tienen los siguientes:

- Ambiente gerencial
- Ambiente del proyecto
- Ambiente de desarrollo

En cada uno de estos ambientes se encuentran las gestiones de cuatro perspectivas distintas:

- **La gestión de tareas:** Enfocado en buscar los incidentes que puedan impedir el desarrollo de la realización satisfactoria de las tareas planificadas y cómo estas tareas intervienen en el desarrollo del proyecto.
- **La gestión de la estructura:** Se refiere a los canales de comunicación determinados por la estructura jerárquica de la organización, en base a la rendición de cuentas propiamente. Es importante comunicar de manera efectiva, generar reportes de omisiones y errores, como también la recompensa en los casos que amerite.
- **La gestión de los actores:** Referida a la acción deliberada frente a los riesgos. Generalmente está representado por el gerente del proyecto, pero también involucra a los demás stakeholders. Todos forman parte proactiva de la gestión en cierta manera.
- **La gestión de la tecnología:** Se centra en identificar los riesgos, componer perfiles de riesgos y evaluar su importancia. Son representaciones de las mejores prácticas de la gestión de riesgos. Se suelen ignorar este tipo de riesgos referenciados a los factores externos, las presiones del medio, omisiones inocentes de tópicos no considerados riesgosos, actores relacionados con experiencias pasadas y otros.

A continuación se presenta el modelo de manera gráfica para una mejor comprensión visual y de esta manera observar la interacción que fluye entre los distintos componentes que lo conforman:

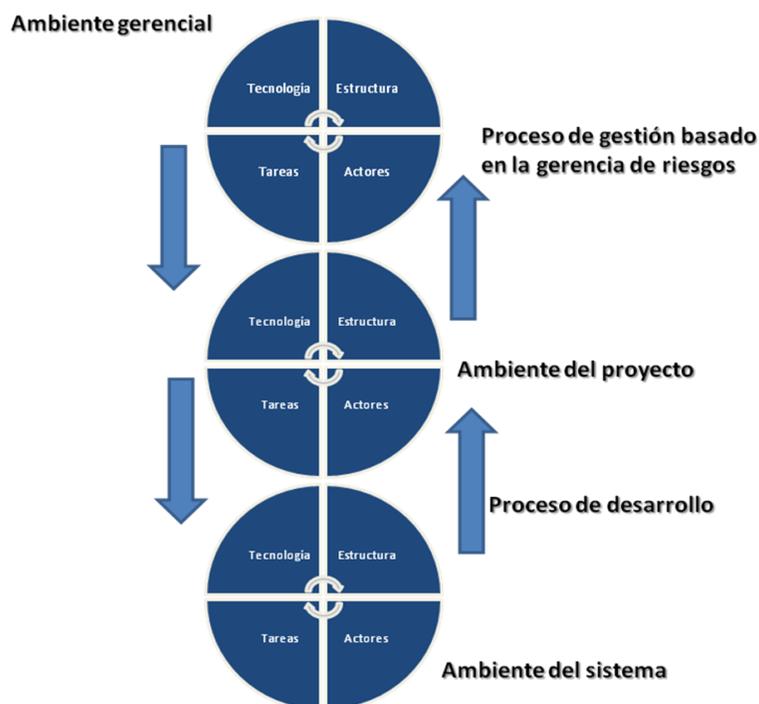


Figura 4. Gestión de actores y ambientes (Evans y Galorath 2006)

La interrelación de estas perspectivas con respecto a los ambientes en que se pueden contextualizar, permiten que este entorno de trabajo para la gerencia de riesgos sea en cierta manera adaptable a cualquier proyecto de software, y permita desarrollarse de manera modular y automatizada en la implementación del sistema de gestión de riesgos de proyectos de software que se desea diseñar.

Los proyectos de software no podían escapar al hecho de ser contemplados bajo la gerencia de riesgos debido a su naturaleza altamente riesgosa lógicamente. Schulte (2004), hace alusión a que existen tres tipos de riesgos que se pueden presentar en proyectos tecnológicos, en los cuales también se encuentran los de desarrollo de software y poseen estas características:

- **Los riesgos propios del proyecto:** Esta clase de riesgos van en contra de cualquier posibilidad de éxito en la realización del proyecto. Se caracterizan

por ser condiciones emergentes o eventos que pueden dañar los entregables propuestos, alterar el presupuesto o el tiempo programado.

- **Los riesgos de los beneficiarios:** Son aquellos riesgos que ocasionan un trastorno en las actividades de los beneficiarios del proyecto, estos pueden generar costos que el beneficiario debe absorber sin indemnización debido a la no contemplación de riesgos de este tipo.
- **Los riesgos corporativos:** Integridad de la data, seguridad, protección antivirus son riesgos a ser tomados en cuenta y que deberían estar en toda lista de chequeo de cualquier tipo de proyecto en la organización.

Indefectiblemente el advenimiento de cualquiera de este tipo de riesgos ha suscitado en gran parte a lo largo de los años que el desarrollo de software sea una tarea altamente riesgosa, tal como los deportes extremos. Es así el caso, que la gerencia de riesgos en proyectos de software se sub-clasifican en niveles más especializados de incidencia que los definidos por Schulte (2004), los cuales son presentados por la Universitat de les Illes Balears (“Curso de ingeniería de software III”) (1999), en base a la existencia de dos tipos de riesgos en la gerencia de los riesgos en proyectos de software, estos son los siguientes:

- **Los de tipo genéricos:** los cuales son comunes a todos los proyectos.
- **Los de tipo específicos:** estos implican un conocimiento profundo del proyecto.

El proceso de identificación de riesgos, es sin duda alguna, el proceso principal de la gerencia de los riesgos antes de avocarse a lo que representa el desarrollo de software en cualquier proyecto. Esto considerado en gran parte a que a continuación se presenta una clasificación basada en una lista de chequeo, en

donde se presentan los más suscitados riesgos no previstos generalmente en proyectos de esta naturaleza y en la mayoría de los casos son atacados de manera reactiva.

Tal es la importancia de esta clasificación, que a continuación se cree realmente necesario presentar cuáles son estos riesgos genéricos, de acuerdo a la clasificación anterior que tanto intervienen y desvían el curso de vida de un proyecto de software. No nos olvidemos de lo que señala Masticola (2007), al referirse que los proyectos de software se materializan eficientemente en términos de cumplimiento de las expectativas de los stakeholders en un 29% y he ahí la importancia de saber lidiar con ellos en el momento oportuno.

A continuación la categorización de los riesgos en base a su identificación dentro de un proyecto de software:

- ***Riesgos relacionados con el tamaño del producto:***
 - Tamaño estimado del proyecto (Líneas de Código/Características del Programa).
 - Confianza en la estimación.
 - Numero de programas, archivos y transacciones.
 - Tamaño relativo al resto de proyectos.
 - Tamaño de la base de datos.
 - Número de usuarios.
 - Número de cambios de requerimientos previstos antes y después de la entrega.
 - Cantidad de software reutilizado.

- ***Riesgos relacionados con el impacto en la organización:***

- Efecto del producto en la cifra de ventas.
- Visibilidad desde la dirección de la organización.
- Fecha límite de entrega razonable.
- Número de clientes que usarán el producto.
- Numero de productos con los que deberá interaccionar.
- Sofisticación del usuario final.
- Cantidad y calidad de la documentación a entregar al cliente.
- Límites legales y gubernamentales.
- Costes asociados al retraso en la entrega.
- Costes asociados a errores en el producto.

- ***Riesgos relacionados con el tipo de cliente:***

- Hay experiencias anteriores con dicho cliente.
- Tiene una idea clara de lo que precisa.
- Está dispuesto a dedicar tiempo en la especificación formal de requerimientos.
- Está dispuesto a relacionarse de forma ágil con el equipo de desarrollo.
- Está dispuesto a participar en las revisiones.
- Es un usuario experto.
- Dejará trabajar al equipo de desarrollo sin dar consejos de experto informático.
- Entiende el ciclo de vida de una aplicación.

- ***Riesgos relacionados con la definición del proceso de producción:***
 - Hay una política clara de normalización y seguimiento de una metodología.
 - Existe una metodología escrita para el proyecto.
 - Se ha utilizado en otros proyectos.
 - Están los gestores y desarrolladores formados.
 - Conoce todo el mundo los estándares.
 - Existen plantillas y modelos para todos los documentos resultado del proceso.
 - Se aplican revisiones técnicas de la especificación de requerimientos diseño y codificación.
 - Se aplican revisiones técnicas de los procedimientos de revisión y prueba.
 - Se documentan los resultados de las revisiones técnicas.
 - Hay algún mecanismo para asegurar que un proceso de desarrollo sigue los estándares.
 - Se realiza gestión de la configuración.
 - Hay mecanismos para controlar los cambios en los requerimientos que tienen impacto en el software.
 - Se documenta suficientemente cada subcontrato.
 - Se ha habilitado y se siguen mecanismos de seguimiento y evaluación técnica de cada subcontrato.
 - Se dispone de técnicas de especificación de aplicaciones para facilitar la comunicación con el cliente.
 - Se usan métodos específicos para análisis de software.
 - Se utiliza un método específico para el diseño arquitectónico y de datos.
 - Está el 90% del código en lenguajes de alto nivel.
 - Hay estándar de documentación de código.

- Se usan métodos específicos para el diseño de pruebas.
- Se utilizan herramientas para llevar a cabo la planificación y control.
- Se utiliza software de gestión de configuraciones para controlar y seguir las actividades a lo largo del proceso.
- Se utilizan herramientas para soportar el análisis y el diseño.
- Se utilizan herramientas de creación de prototipos.
- Se utilizan herramientas para soportar la fase de pruebas.
- Se utilizan herramientas para la generación y mantenimiento de la documentación.
- Se disponen métricas de calidad para todos los proyectos de software.
- Se disponen de métricas de productividad.

- ***Riesgos relacionados con el entorno de desarrollo:***

- Hay herramientas de gestión de proyectos.
- Hay herramientas de gestión del proceso de desarrollo.
- Hay herramientas de análisis y diseño.
- Hay generadores de código apropiados para la aplicación.
- Hay herramientas de prueba apropiadas.
- Hay herramientas de gestión de configuración apropiadas.
- Se hace uso de una base de datos o repositorio centralizado.
- Están todas las herramientas de desarrollo integradas.
- Se ha proporcionado formación a todos los miembros del equipo de desarrollo.
- Hay expertos a los cuales solicitar ayuda acerca de las herramientas.
- Hay ayuda en línea y documentación disponible.

- ***Riesgos relacionados con la tecnología:***
 - Se trata de una tecnología nueva en la organización.
 - Se requieren nuevos algoritmos o tecnología de I/O (Entrada y Salida).
 - Se debe interactuar con hardware nuevo.
 - Se debe interactuar con software que no ha sido probado.
 - Se debe interactuar con un manejador de base de datos cuya funcionalidad y rendimiento no ha sido probada.
 - Es requerida una interfase de usuario especializada.
 - Se necesitan componentes de programa radicalmente diferentes a los hasta ahora desarrollados.
 - Se deben utilizar métodos nuevos de análisis, diseño o pruebas.
 - Se deben utilizar métodos de desarrollo no habituales, tales como métodos formales, Inteligencia Artificial o redes neuronales.
 - Se aplican requisitos de rendimiento especialmente estrictos.
 - Existen dudas de que el proyecto sea realizable.

- ***Riesgos relacionados con la experiencia y tamaño del equipo:***
 - Es el mejor personal disponible.
 - Tienen los miembros las técnicas apropiadas.
 - Hay suficiente gente disponible.
 - Está el personal comprometido en toda la duración del proyecto.
 - Habrá parte del personal dedicado solamente en parte al proyecto.
 - Tiene el personal las expectativas correctas del trabajo.
 - Tiene el personal la necesaria formación.
 - Puede la rotación del personal perjudicar el proceso de desarrollo.

Se puede apreciar que las variables que juegan al momento de combinarse en la aparición de riesgos es notablemente alta en proyectos de esta índole y por tal motivo es necesario idear los mecanismos necesarios para contar con herramientas aplicables a este tipo de proyectos, las cuales cuenten con las características para brindar tanto al gerente del proyecto de software como a los demás stakeholders, una visión clara de las implicancias del desarrollo del proyecto en función de la obtención del producto y todo lo que esto acarrea en términos de esfuerzo, recursos y tiempo.

De esta forma, se conseguiría aglomerar todos los posibles riesgos que puedan afectar al proyecto, para que estos sean canalizados de manera proactiva mediante la automatización de estas tareas en función de un marco tecnológico bien desarrollado y contextualizado producto de la investigación que desea esbozar, la cual deberá contar con todo lo anteriormente expuesto en cuanto a la manera en cómo identificar los riesgos, cómo responder ante ellos, cómo analizarlos, la utilización del marco de trabajo señalado y otra serie de actividades relacionadas, que permitan una gestión del riesgo de manera efectiva y oportuna en proyectos de desarrollo de software.

La manera de como se deben canalizar los riesgos, las mejores prácticas y el gran número de actividades que comprenden este proceso, han sido bien explotadas en gran parte por numerosos autores en la literatura de gestión de los riesgos en proyectos de software. El desarrollo de un sistema de información dirigido a gestionar de manera automatizada tal actividad, ha sido objeto de construcción en gran medida por casas de software que sintieron de alguna manera la necesidad de prestar soporte a esta área de la gerencia de proyectos de una forma u otra, bajo distintos enfoques como se podrá apreciar.

La siguiente figura, proporciona la información recolectada de los distintos productos de software, destinados a gestionar los riesgos en proyectos:

Software	Fabricante	Descripción	Plataforma
SERIM “Software Engineering Risk Management”	Dale Walter Karolak	Software de gestión de riesgos, es el único dirigido especialmente a proyectos de software.	WIN32
RiskTrak	Risk Services & Technology	Permite la identificación de riesgos mediante una amplia base de datos	WIN32
Technical Risk Identification and Mitigation System (TRIMS)	Best Manufacturing Practices	Emplea ingeniería de conocimientos, que se enfocada en la identificación y medición de riesgos técnicos de proyectos.	WIN32
WelcomRisk	Welcome	Brinda una solución para la Identificación Sistemática de Riesgos mediante la utilización de bibliotecas configurables de categorías de riesgos	WIN32
Active Risk Manager (ARM)	Strategic Thought	Es una solución para la Identificación sistemática de riesgos mediante la utilización de bibliotecas configurables de categorías de riesgos.	WEB
Risk-Guide	Jakub Miler, Janusz Górski	Desarrollo de un sistema colaborativo gerencia de riesgos.	WIN32

Tabla 1. Software destinado a la gerencia de riesgos.

La mayoría de estos sistemas de información están destinados como es observable, a la identificación de riesgos basados en listas de chequeo del mismo tipo y con características similares a la clasificación presentada. Se sustentan en fuentes de conocimiento que permiten clasificarlos de manera que el usuario pueda ir viendo como se presentan dentro de lo que es la estructura desagregada de riesgos o también están orientadas a compañías que poseen una amplia base de datos organizacional que les permite generar información de categorías propias de riesgos para de esta manera ir salvaguardando el proyecto y asegurando su planificada continuidad sin alteraciones imprevistas.

El objeto de estudio, producto de esta investigación, propone diseñar conceptualmente un sistema de información que se adapte a las nuevas necesidades de la gerencia de riesgos en proyectos de software, cuestión que no es atacada por los sistemas descritos anteriormente ya que parten de la base de identificar riesgos de proyectos de cualquier índole, esto logrado de una manera fácil, dinámica y rápida, no solamente viéndolo desde un punto de vista desde la perspectiva del gerente del proyecto, sino involucrando a todos los stakeholders

del mismo, de manera de crear un ambiente de trabajo colaborativo virtual para la gerencia de riesgos del proyecto, en el cual se permita visualizar la mayoría de los riesgos latentes en un proyecto a manera de indicadores y permitir al usuario conocer cuáles serían las mejores prácticas para solventar o enfrentar tal situación generada, basándose en el instrumento de las listas de chequeos mostradas anteriormente, conjuntamente con el instrumento desarrollado por López (2007), el cual permite la evaluación de la gestión de riesgos en proyectos sustentándose en los distintos procesos que conforman dicha área de conocimiento que señala el PMI (2004).

Este instrumento señalado permite determinar de manera clara y precisa, a través de escalas valorativas, el nivel en el cual se encuentra cada proceso separado en sí mismo, a través de los tres aspectos fundamentales que lo determinan (entrada, herramientas/técnicas, salidas) para de esta manera saber si el proyecto es viable en términos de gestión de los riesgos o no. Cabe destacar que este instrumento estará enfocado esencialmente a proyectos de software, por lo cual los aspectos mencionados de cada proceso estarán delimitados partiendo de la base de su aplicación a este tipo de proyectos.

Anudado a este planteamiento la herramienta debería contar con las siguientes características:

- **Multiplataforma:** El sistema de información podrá ser instalado en cualquier sistema operativo, para de esta manera ser altamente adaptable a cualquier entorno corporativo.
- **Presentación automática de riesgos comunes:** Todo proyecto posee riesgos genéricos, por lo tanto serán presentados de manera de que el usuario pueda interpretarlos y asociarlos al proyecto de manera rápida y sencilla.

- **Interfaz amigable:** El sistema de información será completamente amigable e intuitivo en la realización de tareas comunes, adaptándose al tipo de usuario que se definirá de acuerdo al uso que se le vaya a dar y las posibilidades de acción que tendrá dentro del mismo (implementación de roles y seguridad de acceso a información).
- **Generación de reportes:** Imprescindibles para el manejo de información en cualquier momento sobre el estatus de los riesgos durante cualquier etapa del ciclo evolutivo de la gestión de los riesgos de proyectos de este tipo.
- **Soporte de multilinguaje:** La interfaz de usuario permitirá realizar el cambio de idiomas, dependiendo de la localidad del usuario.
- **Desarrollado en tecnología Web:** El sistema de información estará desarrollado en ambiente Web, con la finalidad de que los stakeholders del proyecto, puedan ingresar y monitorear el estatus de la gestión de riesgos, desde cualquier ubicación y en cualquier momento.
- **Soporte de consulta de proyectos anteriores:** Este modulo permitiría al usuario consultar a manera de historial, la forma en cómo se identificaron los riesgos para proyectos anteriores y como fue su evolución en cuanto a comportamiento durante el ciclo del proyecto.
- **Diagnostico de la gestión de riesgos:** Basándose en el instrumento de López (2007), diagnosticar mediante gráficos e indicadores el estatus de la gestión de riesgos del proyecto.
- **Creación de tablas de riesgos basadas en los riesgos previamente seleccionados:** Permitirán categorizar los riesgos, de manera de medir el impacto y la probabilidad de ocurrencia dependiendo.

- **Manejo de múltiples proyectos:** La gerencia de riesgos no deberá estar delimitada a un solo proyecto por vez, sino que se podrán ver y acceder a la información de múltiples proyectos en curso.
- **Permitir foros de discusión de proyectos:** Los distintos stakeholders podrán opinar y discutir los riesgos identificados en un proyecto determinado, para de esta manera aportar soluciones para su tratamiento.
- **Simulación de escenarios de riesgos:** Ver como determinados riesgos puedan influir en el ciclo de vida del proyecto y como sería el comportamiento del proyecto en términos de desviación y escenarios fortuitos.
- **Poseer 4 grandes áreas de trabajo:** Las cuales van a estar referidas al marco de trabajo seleccionado el cual se basa en la gestión de los actores, la gestión de las tareas, la gestión de la tecnología y la gestión de la estructura. De esta manera permitiría al usuario segmentar los riesgos de manera independiente y trabajarlos por separado para un mejor aislamiento y tratamiento de los mismos.
- **Análisis cualitativos y cuantitativos de riesgos:** Una vez generada la identificación de riesgos, el sistema debería arrojar reportes que permitan mostrar este tipo de información a fines de informar al usuario las acciones posibles a tomar en caso de valorar cualquier riesgo latente en el proyecto.
- **Independiente de la plataforma de acceso:** Este punto hace referencia a que la herramienta debe poseer la funcionalidad de permitir que pueda ser accedida tanto por computadores de escritorios (PC), como dispositivos móviles, a fin de darles libertad a los stakeholders a la hora de hacer gerencia de riesgos del proyecto en forma remota.

Cabe destacar que los aspectos citados son a grandes rasgos las principales funcionalidades con las cuales se estaría implementando el diseño del software de gestión de riesgos en proyectos de software. Estas son variables en el tiempo y considerables a modificación en base a nuevas investigaciones, por lo que no es un producto cerrado bajo ningún concepto.

Para que el lector tenga una mejor comprensión del alcance final de esta herramienta de software se presenta de forma grafica la conceptualización de la misma bajo un aspecto funcional, tomando como referencia el punto de vista del usuario y las acciones que este podría desarrollar en ella una vez desarrollado el producto.

En el diagrama se pueden apreciar 2 áreas principales, una que comprende las tareas o acciones básicas con las cuales el sistema se sustenta y/o alimenta principalmente y la otra parte viene dada por las acciones o tareas especializadas que el usuario puede ejecutar, que no son más que las tareas que derivan en la gestión de los riesgos de los proyectos que se pudieran estar manejando en un momento determinado.

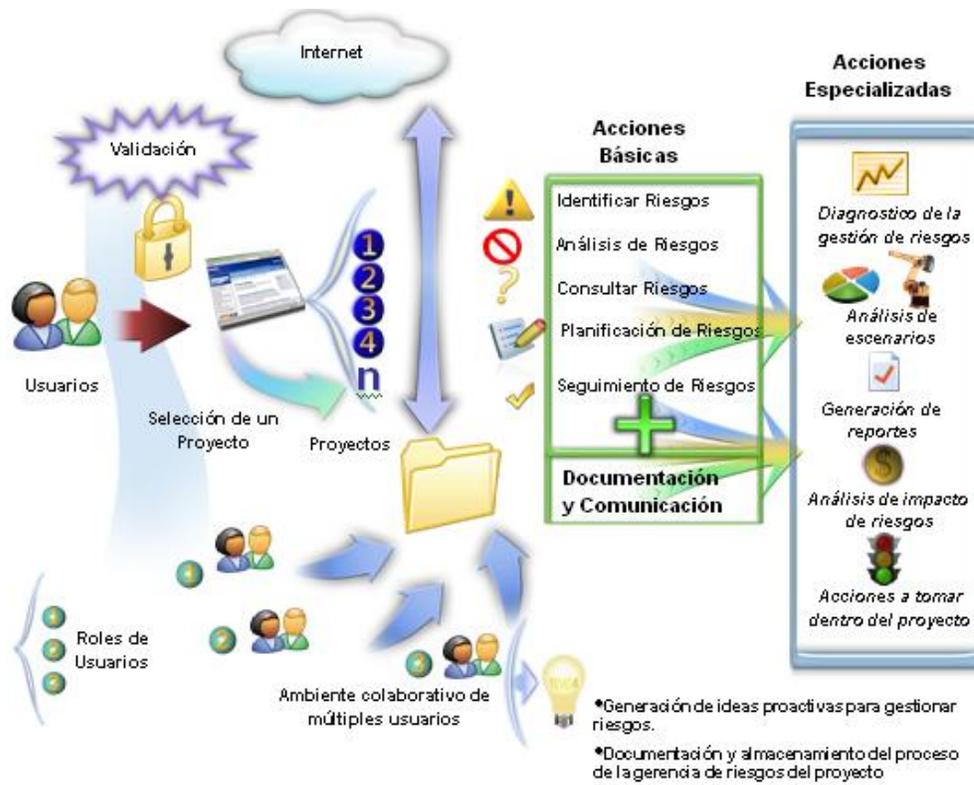


Figura 5. Diseño conceptual de la herramienta de software

CAPÍTULO III

MARCO METODOLOGICO

1. Breve introducción

Ante la necesidad de contar con un marco de trabajo claro para la investigación, el cual permita el desarrollo de la herramienta que se ha definido conceptualmente, es necesario contextualizar en donde se encuentra ubicada y el tipo de investigación que es, para de esta forma pueda ser desarrollada, en virtud a la naturaleza misma de ser proyecto de software.

Esto se debe en particular a que esta investigación se encuentra enfocada a la obtención de una creación tangible que puede ser utilizada como solución a un problema en específico, el cual fue ya planteado con anterioridad en el marco conceptual de esta investigación. Al poseer este tipo de características se le considera como un proyecto especial. Este tipo de proyectos se caracterizan por su carácter innovador que puede producir un aporte significativo al conocimiento sobre el tema seleccionado siempre y cuando se fundamente de manera correcta la parte teórica de la investigación y la descripción de la metodología utilizada.

A tales fines se presentara una metodología de investigación que se focaliza en el desarrollo de la herramienta, permitiendo de manera análoga ver los productos resultantes de cada una de las etapas que se derivan de su realización, para de esta manera generar un valor agregado que permita tanto al lector, como al diseñador y desarrollador de la herramienta, tener el conocimiento general y global de cómo desarrollar el proyecto.

2. Tipo de investigación

El presente estudio de investigación posee las siguientes características:

- **Aplicada:** También conocida como práctica o empírica. Se basa en la aplicación o utilización de los conocimientos adquiridos. Este tipo de investigación depende estrechamente de los resultados y avances obtenidos, fundamentados siempre en su marco teórico. La importancia de este tipo de investigación son sus consecuencias prácticas, producto del objeto de estudio.
- **Factible:** Producto de su misma naturaleza, ya que esta investigación se encuentra asociada a un contexto organizacional de carácter interno, como es la gestión de riesgos en proyectos, específicamente de software. Su enfoque está concebido en el hecho de ser un producto el cual pueda brindar una utilidad específica a los gerentes de proyectos avocados a esta rama de la tecnología y servir de herramienta que ayude y derive en una mejor gestión en el área de proyectos a través del uso e implementación de las mejores prácticas en materia de riesgo.

3. Estrategias para la Recolección, Procesamiento y Análisis de la Información

En virtud de las características de esta investigación, se determinó necesaria la conceptualización de un marco de investigación capaz de acoplarse a lo que significa el desarrollo de la herramienta a nivel de ingeniería de software en toda su magnitud enfocado desde el punto de vista de investigación.

De esta forma se podrá conocer como estará estructurada la secuencia lógica que permitirá desarrollar la herramienta, a fines de contar con un marco sólido de trabajo, sustentando en metodologías claras y definidas para la consecución de los objetivos planteados.

Para el desarrollo de la investigación se opto por dirigir su estructura a como Marcos (2002) representa el enfoque que caracteriza el método de investigación en la ingeniería de software, basándose principalmente en los pasos descritos por Bunge (1978), los cuales se detallarán seguir en toda investigación científica.

Con este enfoque se puede apreciar el paralelismo existente entre los pasos descritos basados en el método hipotético-deductivo, con los métodos de desarrollo de software, los cuales se detallaran más adelante. Para ejemplificar dicha estructura a continuación se muestra la figura 6 la cual detalla el método de investigación empleado para el desarrollo del software.

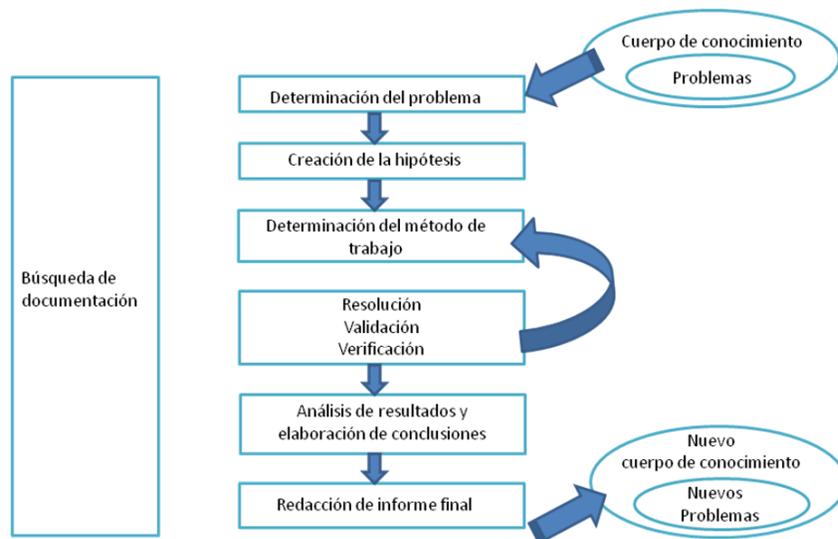


Figura 6. Métodos de investigación. (Marcos, 2002)

Para tener un poco más de conocimiento en relación a cada uno de estos pasos, a continuación se definirán brevemente cada uno de ellos, a fin de estructurar metodológicamente la manera en cómo se piensa desarrollar el proyecto de tales características.

- ***Etapa 0: Búsqueda de información:*** Enfocado particularmente en toda la contextualización de la información a ser procesada para la elaboración del objeto de estudio. Abarca el dominio del producto a desarrollar, herramientas disponibles para el desarrollo del producto y todo aquello que proporcione de cierta manera algún contenido sustancial de información que permita llevar a cabo lo que se desea. Una vez claro este punto, se tendría como producto de esta etapa las herramientas que permitan alinear el desarrollo del proyecto en dirección a lo que es el propio alcance del proyecto.

- ***Etapa 1: Determinación del problema:*** Definición clara y objetiva del problema a resolver. Esta etapa se asemeja considerablemente a la etapa de captura de requisitos en el proceso de desarrollo de software.

La ingeniería de requisitos es una tarea clave en lo que respecta al diseño y desarrollo futuro del software, por lo que la mayor claridad que se tenga en las necesidades de usuario y el problema planteado determinara en gran parte la viabilidad y éxito del proyecto.

- ***Etapa 2: Creación de hipótesis:*** Para señalar este punto es necesario expresar lo que Marcos (2002) cita a continuación:

“En los métodos tradicionales de investigación científica la hipótesis se formula en términos causales. Estas hipótesis son conjeturas de hechos que el método científico deberá contrastar y verificar. Sin embargo, es fácil comprobar que la hipótesis en una investigación en

ciencias de la ingeniería de software no responde a un planteamiento de causa-efecto. Recuérdese que el objeto de estudio en este tipo de ciencia es la construcción de nuevos objetos (modelos, técnicas o, como en nuestro caso de estudio, métodos), que, por no existir, no son susceptible de experimentación.” (p. 9).

Partiendo de este punto de vista, la hipótesis se formulará como la descripción del nuevo objeto que se va a construir, en términos de características, funcionamiento, requisitos y uso del mismo por parte del usuario al cual va enfocado, lo cual ha quedado explícito y desarrollado en el marco conceptual de la investigación.

De manera implícita se deja claro la necesidad del estudio, la cual es en sí, la carencia de una herramienta con tales características ya descritas y su propia necesidad de elaboración, la cual radica en la razón del estudio de investigación propiamente.

- **Etapas 3: Definición del método de trabajo:** En virtud de que los métodos de investigación no definen claramente el método a seguir para la verificación y resolución del problema, estos se definirán en función a cómo se comporta el método de desarrollo de software.

Por tal motivo la retroalimentación es algo vital, lo cual es producto de las tres actividades representadas por la resolución, validación y verificación, cada una de estas son tareas base, a la hora de implementar un desarrollo de software. Esta tarea de por sí es un producto muy fluctuante a los cambios dentro de su propio alcance y necesita ser validado de manera constante.

El desarrollar un producto de software a partir de un modelo conceptual concebido a través de las técnicas de ingeniería de requerimientos, debe ir de la mano de una metodología clara y sustentada en patrones de composición y funcionamiento que auguren su éxito al momento de su aplicación.

Para el caso de esta investigación, los patrones de arquitectura a utilizar vendrán determinados desde la base del mismo diseño, que luego derivará en su implementación propiamente como producto, para de esta manera homogenizar la puesta en marcha de la construcción del mismo y de esta manera contar un marco de trabajo que disponga de las posibilidades de llevar el desarrollo del producto sin mayores dificultades y contratiempos.

Para conocer que es la palabra arquitectura dentro lo que es la construcción de software es necesario citar a IEEE (2006) la cual expresa que este concepto va referido a la organización fundamental de un sistema (en nuestro caso software), comprendiendo sus componentes, las relaciones entre ellos y el ambiente, y los principios que gobiernan su diseño y evolución.

Partiendo de este concepto se especifica el contar con un marco de desarrollo de un sistema orientado a objetos completamente, el contar con una amplia biblioteca de clases de funciones, que permitan la realización de tareas comunes dentro del software a diseñar, la independencia del lenguaje (dependiendo de la tecnología a seleccionarse para la implementación).

El definir la arquitectura básica del software determinará en la especificación y generación de los componentes de negocio y de esta

manera poder establecer el mapa de navegación hacia el establecimiento de un entorno de desarrollo orientado a componentes que coexistan unos con otros dentro de la aplicación para dar respuesta a las necesidades detectadas en la ingeniería de requerimientos.

El marco de trabajo llega hasta el momento en el cual el software esté listo y se hayan completado las últimas pruebas de usuario y calidad de funcionamiento, a fines de garantizar el comportamiento estable del mismo.

- **Etapa 4: Resolución, validación y verificación:** La resolución se centra en definir la especificación del desarrollo del software y también las técnicas y actividades implementadas para conseguir su propio desarrollo. Para esto es necesario cubrir el análisis de casos de estudios y la misma imaginación y creatividad que dan vida al producto.

La validación va ir dada en la implementación de prototipos a fines de ir limando los requerimientos funcionales que se desean concretar, tanto en la interfaz como en el funcionamiento del software, cerrando con la validación, la cual no es más que la aplicación de los casos de prueba, pruebas de stress y otra serie de acciones que permitan ver la respuesta del software ante la interacción del usuario bajo distintas situaciones.

La fase de resolución se resumiría a la creación de la hipótesis en un método empírico, lo cual determinaría en la validación del método utilizado para la verificación de los resultados empleado en el desarrollo del software propiamente.

La implementación de esta metodología de trabajo para alcanzar el objetivo planteado de la investigación estará sustentada en la utilización de la “programación extrema (XP)”, para lo que es el desarrollo y construcción del software propiamente.

Es una necesidad de contar con una metodología de desarrollo de a la hora de emprender el desarrollo de un producto. En virtud de la gran cantidad de enfoques y metodologías que existen en la actualidad se evidencio como justificada la selección de la misma a raíz de una serie de aspectos que reúne esta metodología y que se citan a continuación según Calero (2003):

- XP define cuatro variables para proyectos de software: coste, tiempo, calidad y ámbito.
- Además de estas cuatro variables, se propone que sólo tres puedan ser establecidas por las fuerzas externas (jefes de proyecto y clientes), mientras que el valor de la cuarta variable debe ser establecido por los programadores en función de las otras tres.
- Se centra en 4 valores fundamentales para el desarrollo del trabajo en función del factor cambio, el cual es algo de lo que no se puede escapar en la actividad del desarrollo del software debido a motivos tales como: cambiarán los requisitos, las reglas de negocio, el personal involucrado en el desarrollo, la tecnología, todo es susceptible a cambiar en un momento dado, y cada una de estos valores van enfocados al comportamiento de cada una de las partes involucradas en el desarrollo del software propiamente, estos valores son:

- **Comunicación.**
 - **Sencillez.**
 - **Retroalimentación.**
 - **Valentía.**
- Luego de tener claro los valores esta metodología se enmarca en las tareas que se deben llevar a cabo para completar lo esperado y están representadas de esta manera:
 - **Codificar** (realizar los componentes de negocio, interfaz de usuario y sistema en general).
 - **Hacer pruebas.**
 - **Escuchar** (en función de lo que se plantea el usuario del sistema).
 - **Diseñar** (estructurar la lógica del sistema de manera coherente).
- Una vez expuesta la escena en cómo se deben hacer las cosas, la metodología XP canaliza su accionar en una serie de tareas que se detallan lo que es la solución del propio sistema y se señalan a continuación:
 - Una planificación entre las partes involucradas del desarrollo del sistema. Aquí se toman en cuenta aspectos tales como: el ámbito, prioridad, composición de versiones, fechas de versiones, estimaciones, consecuencias, procesos y programación detallada.
 - La realización de lo que se llaman “pequeñas versiones” operativas del sistema.
 - El diseño, el cual se basa en la metáfora, la cual se centra en historias de los participantes de cómo funciona el sistema y que opinan del mismo en ciertos aspectos relevantes a considerar. Un diseño sencillo, y por último un desarrollo. Este desarrollo va contemplado en una serie de actividades tales

como la recodificación, la programación por parejas, la propiedad colectiva, una integración colectiva y algo que se llama “40 horas semanales” lo cual se enfoca a que la motivación va en función del grado de cansancio del programador y establece una serie de recomendaciones al respecto. También se maneja el tema del Cliente In-situ, lo cual como su nombre lo dice, es necesario trabajar directamente con las personas que utilizaran el producto y no de manera aislada. Es necesario también contar con estándares de codificación que permitan una homogeneidad de lo que se está haciendo de cierta manera pre-establecida.

- La realización de las pruebas.
- El cómo va a funcionar el software. Sustentándose en las mismas premisas señaladas en el apartado del diseño.

De esta forma se logra contar con un marco de trabajo claro para emprender el desarrollo de la investigación y de esta manera canalizar el esfuerzo en una dirección marcada de una forma clara sustentada en la experiencia.

- **Etapa 5: Análisis de resultados y elaboración de conclusiones:** Simplemente se enfoca a recapitular si lo que se planteo inicialmente como hipótesis al inicio de la investigación fue conseguido a través de los resultados del producto obtenido.

La determinación de los puntos no completados y nuevos enfoques surgidos del problema original son de gran importancia que sean tomados en cuenta para la retroalimentación de la misma investigación y futuras investigaciones sobre el tema.

- **Etapas 6: Redacción del informe final:** Toda la documentación relativa a cada una de las etapas que se han citado previamente son la fuente de este informe. Esto a fines de generar una fuente de conocimientos, que permite obtener lo más importante del producto de la investigación, sus limitaciones y sus principales objetivos.

La realización de manuales y requisitos de funcionamiento son parte a desarrollar en todo software por lo que en esta etapa se contempla de manera directa la recopilación de la información de uso del mismo a fines de plasmar su funcionamiento de manera simple y acorde a lo que se esperaba desde el planteamiento de su propio alcance.

A fines ilustrativos la siguiente figura recapitula toda la información expuesta, en función de la analogía entre los métodos de investigación enfocados a la ingeniería del software vs el método de desarrollo de software propiamente.

Etapas	Método de Investigación enfocado a la ingeniería de software	Método de desarrollo software
E0: Documentación	Problema a resolver método de investigación	Dominio de la aplicación metodologías, técnicas, herramientas, etc.
E1: Determinación problema	Estudio de campo Delimitación del problema	Análisis de dominio Captura de requisitos
E2: Creación hipótesis	Descripción del objeto a construir	Especificación de requisitos
E3: Definición de Método	Selección paradigma (cualitativo, cuantitativo, etc.). Selección del método (experimental, investigación en acción, etc.). Adaptación al problema (técnicas de experimentación, marcos de validación, etc.)	Selección paradigma (estructurado, OO, etc.) Selección del método (Métrica, XP, etc.) Adaptación al problema (técnicas que se usarán, notaciones, etc.)
E4: Resolución, Validación, Verificación, Resolución, V&V	Casos de estudio, creatividad Casos de prueba, Prototipo herramienta, Investigación en acción	Casos de estudio, creatividad Juego de pruebas, Prototipo, Programación extrema XP
E5: Análisis	Contrastación hipótesis	Contrastación requisitos de usuario
E6: Informe final	Hipótesis, Método, Conclusiones, etc.	Requisitos, Manual de usuario, etc.

TABLA 2. Métodos de investigación vs. Métodos de desarrollo de software. (Marcos, 2002)

A raíz de lo expuesto se podría decir que con esa información se vislumbra un panorama en donde se puede ver en donde la investigación se encuentra concebida y de cómo se llevara cabo. A tales fines, se utilizará la estructura desagregada de trabajo, la cual permitirá a cierto nivel de detalle las tareas que comprenden la elaboración del objeto de la investigación a fines de conceptualizar la labor necesaria del mismo.

En la figura 7, se podrá observar las diferentes fases que corresponden al desarrollo en del software siguiendo la estructura desagregada de trabajo (EDT) a continuación:

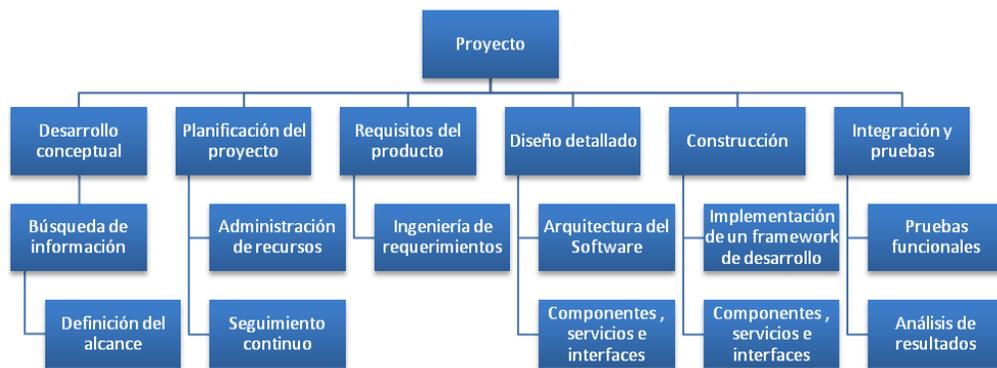


Figura 7. Diagrama EDT del proyecto de software a realizar.

Para representar en el tiempo la ejecución de cada una de estas tareas se presenta a continuación el cronograma de trabajo probable para la consecución del objetivo planteado por la investigación.

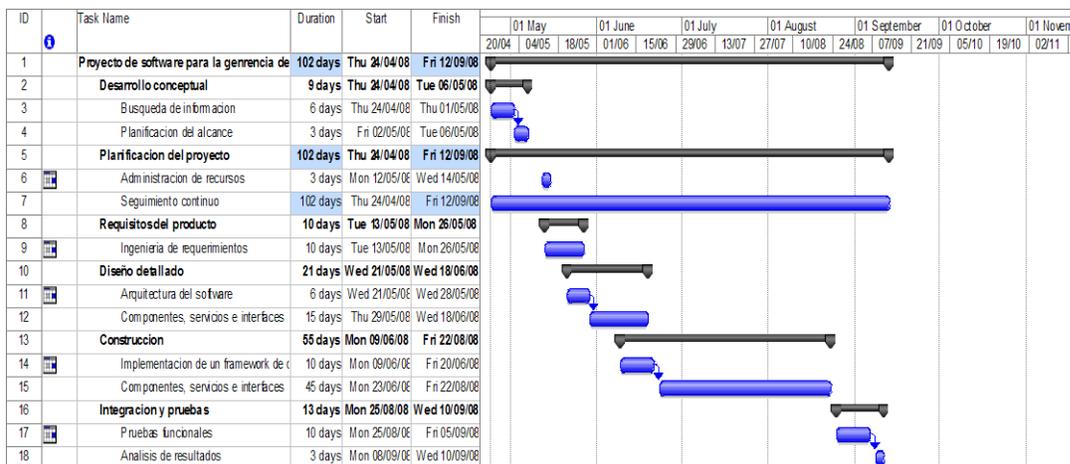


Figura 8. Diagrama Gantt del proyecto.

4. Factibilidad

La factibilidad de la investigación viene dada por el hecho de contar con los recursos y el conocimiento necesario para el desarrollo de la investigación como también la herramienta de software que se ha diseñado conceptualmente producto de esto, a fines de que a través del marco de trabajo propuesto y todo el sustento bibliográfico recopilado, poseer la capacidad de mantener controlada la situación en términos de calidad, costo y tiempo de lo que implicaría la puesta en marcha como proyecto.

Producto de esta factibilidad, el desarrollo de esta herramienta deberá derivar en ciertas cualidades esperadas tales como:

- Definir los riesgos asociados a un proyecto enmarcados en la utilización de tablas de riesgo propuestas por los autores citados en la investigación, permitiendo al usuario la facilidad de adopción de los riesgos más comunes en proyectos de software, dependiendo de las características del proyecto.

- Monitorear dichos riesgos a través del uso de indicadores de gestión, capaces de mostrar el estatus del proyecto en cualquiera de sus sub-procesos.
- Facilitar al usuario la adopción de medidas para la toma de acciones correctivas en la reducción de riesgos dentro de un proyecto de software, basado en experiencias previas “pre-cargadas” dentro del sistema a través del uso de árboles de decisión.
- Permitir interpretaciones analíticas de resultados, producto de la asignación de ponderaciones, creación de métricas, cálculos de impacto de los riesgos identificados y simulación estocástica de baja moral.
- Generar una fuente de conocimientos a partir de la data recolectada por cada uno de los proyectos analizados por la gerencia de riesgos implementada por el software.
- Aprovechar el uso del Internet de manera que el usuario pueda monitorear la operación del sistema y realizar todas las tareas de manejo de riesgos dentro del producto sin importar su ubicación física durante el transcurso del proyecto a través de una arquitectura cliente-servidor.
- Permitir al usuario aportar datos de interés sobre riesgos específicos de manera rápida y clara una vez ya definidos.

La idea central radica en que el modelo está pensado en ser desarrollado para que funcione bajo un ambiente en línea colaborativo de gestión, involucrando a todos los dolientes del proyecto, a fin de no descartar sus opiniones y permitirles exponer sus experiencias y recomendaciones en torno a un riesgo identificado, logrando así la gestión del conocimiento de los riesgos que se presenten.

Por defecto los stakeholders se sentirían mas tomados en cuenta y dueños del proyecto, permitiéndoles manejar el riesgo de una forma positiva o minimizando su impacto lo máximo posible en base a la información que ellos mismos ingresen a la herramienta. Lo principal que se deriva de todo esto es simplemente que aprenderán a “hacer gerencia” con eficiencia en cuanto al manejo de los riesgos de un proyecto de software, realmente sin mucho esfuerzo, bajo una metodología clara y precisa.

CAPÍTULO IV

LECCIONES APRENDIDAS Y RECOMENDACIONES

El desarrollo de la herramienta de software debería estar enmarcado bajo la metodología planteada, la cual ha de enfocarse en la utilización lógica de una arquitectura de software, a fines de brindar la funcionalidad deseada tanto para los desarrolladores en términos de construcción del software como para los usuarios en cuanto al uso de la herramienta.

La conceptualización del diseño se deberá centrar en presentarle al usuario la mayor cantidad de riesgos posibles asociados a una clasificación pertinente del riesgo, tanto los precargados por la herramienta, como también los riesgos suscitados en experiencias anteriores con proyectos de la misma naturaleza o complejidad.

El minimalismo debería ser parte fundamental en aspectos de diseño, a fin de no atiborrar de información innecesaria al usuario y este pueda desplazarse cómodamente al momento de navegar por la herramienta, tanto en un ordenador de escritorio como en un dispositivo móvil.

Utilizar como lecciones aprendidas todos los riesgos que se deriven de un proyecto, de esta manera se propiciara la gerencia de conocimiento dentro de la organización y permitiría a todos los involucrados aprender de los errores y tenerlos en cuenta para futuras ocasiones.

Sería beneficioso proyectar este tipo de experiencias en otros ámbitos de la gerencia de proyectos, especializando sistemas de información que deriven en una mejor gestión de proyectos de software.

Involucrar como usuarios a todos y cada uno de las personas que intervengan en el proyecto a desarrollar, a fines de contar con todos los puntos de vista y así de esta forma propiciar lluvias de ideas, capaces en desembocar en una gran cartera de riesgos a manejar y controlar durante el proyecto, a fines de garantizar su correcta finalización en términos de costo, calidad y tiempo.

Una vez identificados y controlados estos riesgos, convertir estas amenazas en oportunidades para los stakeholders del proyecto haciendo notar que gracias al enfoque proactivo se logra cambiar en sentido positivo los sucesos inesperados que afectan a los proyectos en cualquier momento.

REFERENCIAS BIBLIOGRÁFICAS.

Boehm, B. W (1989). Software project Management: principles and examples. IEEE Trans.

Bruckner, Robert M., List, Beate y Schiefer, Josef. (2004). Risk-Management for Data Warehouse Systems. Proceedings of the Institute of Software Technology. Vienna, Austria

Chatterjee, Dhiman y Ramesh, VC (1999). Real Options for Risk Management in Information Technology Project, Proceedings of the Hawaii International Conference on System Sciences. Chicago, Illinois. (paper).

Graham, Robert y Englund, Randall (1999). Administración de proyectos exitosos. México. Prentice Hall. (p.1)

Higuera, Ronald y Haimes, Yacov (1996). Software Risk Management. Software Engineering Institute Carnegie Mellon University. Pittsburgh, Pennsylvania.

Heerkens, Gary (2001). Project Management. McGraw-Hill.

Hilson, David , (1998). Project Risk Management: Future Developments. www.pmi-muc.de/Vortraege/200606/BestPracticeFutureDevt%20PMI%20df.pdf

Horine, G. M. (2005). Absolute beginner's guide to project management.

IEEE 1540 – Software Engineering Risk Management (2001). Measurement-Based Life Cycle Risk Management. Aspen, Colorado.

IEEE Transactions on Software Engineering (2006). Introduction to the Special Issue on Software Architecture.

ISO 10006:2003 (2003). Quality management systems. Guidelines for quality management in projects

Johnson, Donna L. (2006). Risk Management and the Small Software Project Risk Project. Nashville, Tennessee: LOGOS International, Inc.

Karolak, Dale Walter (1997). Software Engineering Risk Management. Aspen, Colorado.

Lister, De Marco (2003). Risk Management during Requirements. www.systemsguild.com/pdfs/s5req.lo%201.pdf

Lopez, Emmanuel (2007). Gerencia del desempeño. UCAB Caracas, Venezuela.

Lyytinen Kalle, Mathiassen Lars y Ropponen Janne (1996). A Framework For Software Risk Management. Jyväskylä, Finland. Scandinavian Journal of Information Systems.

Masticola, Stephen P. (2007). A Simple Estimate of the Cost of Software Project Failures and the Breakeven Effectiveness of Project Risk Management, Proceedings of the 29th International Conference on Software Engineering Workshops. Princeton NJ

Microsoft (2002). MSF Risk Management Discipline v.1.1. Microsoft Solutions Framework. <http://www.microsoft.com/msf>.

Miler, Jakub y Górski, Janusz (2001). Software support for collaborative risk management. Mielno, Poland.

Palacios, Luis E (2005). Gerencia de Proyectos, Un enfoque latino (3a. ed.). UCAB

Pressman, Roger S (2001). Ingeniería del Software, Un enfoque practico (Quinta Edición). Madrid. España: Editorial Mc. Graw Hill.

Project Management Institute (2004). Guía de los Fundamentos de la Dirección de Proyectos (Guia del PMBOK). (Tercera Edición). PA EEUU.

Real Academia Española (RAE). <http://rae.es/rae.html>

SEI. Software Engineering Institute. (2004).<http://www.sei.cmu.edu/programs/sepm/risk/definition.html>. Página vigente al 07-Sep-2004.

Schulte, Peter (2004). Complex IT Project Management: 16 Steps to Success. (pp.420-421). CRC Press.

Solarte, Leonardo, Montilla Omar, García Mónica (2003). Modelos de gestión de proyectos en cinco (5) organizaciones líderes nacionales. (p.4)

Suri, P.K (Dr.) y Soni, Rachna (2007). Stochastically Simulating Low Morale - A Risk Factor Effect on Risk Management. Kurukshetra.

Suri, P.K (Dr.) y Wadhwa, Manoj (2007). Identification and Assessment of Software Project's Risk. Kurukshetra India.

Taylor, James (2004). Managing Information Technology Project: Applying Project Management Strategies to Software, Hardware, and Integration Initiatives. AMACOM.

The Orange Book. Management of Risk - Principles and Concepts. (October 2004). HM TREASURY.

Universitat de les Illes Balears (1999).Curso de ingeniería de software III. España.

Universidad de Salamanca (1999). Curso de gerencia de riesgos y seguros en la empresa.

Wallmüller, E (Dr.) (2005). Risk Management for IT and Software Projects
[http: //www.itq.ch](http://www.itq.ch), Züric