

Universidad de las Ciencias Informáticas
“Facultad 4 y 6”



Título: “Propuesta para la Gestión de Riesgo
en los proyectos productivos de la UCI”

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

Autores:

Yanisleidy López Cabrera

Tailys Alvarez Lamas

Tutor:

MSc Yamilis Fernández Pérez

Cuidad de La Habana

Julio del 2007

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Tailys Alvarez Lamas
(Autora)

Yanisleidy López Cabrera
(Autora)

MSc Yamilis Fernández Pérez
(Tutora)

DATOS DE CONTACTO

Tutora: Yamilis Fernández Pérez, ciudadana cubana, graduada de Ingeniera en Sistema Automatizado de Dirección, en 1992 en el ISPJAE. Impartió docencia en universidades desde 1992, tres años más tarde logra la categoría de Profesora Asistente. En este mismo año (1995), realiza la maestría en Informática Aplicada. Ha desarrollado trabajos con Universidades extranjeras en Brasil, Bolivia y Canadá. Actualmente es la jefa del Departamento Central de Ingeniería y Gestión de Software de la UCI, desde su fundación, además imparte la asignatura de Ingeniería de Software. Obtuvo el certificado del Curso de Introducción al CMMI que imparte el SEI.

AGRADECIMIENTOS

Agradezco:

- *A mi mamá y mi papá, todo su amor, sus consejos sacrífico y preocupación constante, para que su hija lograra hacer realidad sus sueños lo cual siempre estará en mi corazón.*
- *A mi hermano Javi por su cariño.*
- *A mi querida abuela Amalia por su apoyo, sus consejos y toda su dedicación además todas las cosas buenas que me ha enseñado.*
- *A mi abuelo Pedro Cabrera que aunque no lo puedo tener junto a mí, su recuerdo y su ejemplo estará siempre en mi corazón, quiso que su nieta como los demás nietos fuera una persona preparada y capacitada para enfrentar la vida y así será mi querido abuelo .*
- *A José mi querido novio por su amor, comprensión, dedicación; por estar siempre a mi lado en las cosas buenas y en las malas, por esas y otras razones nunca lo podré olvidar.*
- *A mi abuela Victoria por su cariño y preocupación.*
- *A mis suegros por todo su amor, cariño y atención constante, a ellos que son para mí como padres.*
- *A mis tíos y tías todo su amor, apoyo y preocupación.*
- *A Leopoldina, que siempre me aconsejó y me enseñó el camino correcto en esta vida.*
- *A mi prima Ladita que siempre ha sido para mí un buen ejemplo, a ella y su esposo Leonides, por su preocupación, apoyo y cariño.*
- *A mis primos por su cariño*
- *A toda mi familia por esa preocupación y ayuda los cuales me sirvieron de mucho.*
- *A la familia de José por toda su atención y apoyo.*
- *A mi madrina y familia que siempre me ayudaron y me dieron buenos consejos sobre la vida.*
- *A mis inolvidables amigas Judith, Janny, Suanly, Lesley y Eliane por toda su comprensión, ayuda y todo su cariño.*
- *A mis compañeros que cursaron conmigo los 5 años de Universidad que de una forma u otra colaboraron a este trabajo.*
- *A mi querida compañera de tesis Tayli la cual es una magnífica compañera y amiga, que juntas supimos hacer realidad este hermoso logro y a su novio Arley por todo su apoyo y paciencia.*
- *A todos mis profesores por su enseñanza y todo el conocimiento que me transmitieron que quedaran siempre grabados en mí.*

YANIS

AGRADECIMIENTOS

Agradezco:

A mis padres, por brindarme todo su apoyo, confianza y amor, por creer siempre en mí, por ser mi ejemplo y mi guía a lo largo de todo este trayecto, ustedes se merecen mucho más que esto.

A mis abuelos y bisabuela, aunque ella no se encuentre entre nosotros, siempre depositaron su confianza en mí y de una forma u otra contribuyeron en mi formación.

A mi novio Arley, por estar siempre a mi lado, aconsejarme en los buenos y malos momentos, por ayudarme en el desarrollo de este Trabajo de Diploma, por guiarme siempre hacia el camino correcto, por tener tanta confianza y paciencia conmigo.

A mis suegros, por haberme ayudado a transitar a lo largo de este periodo tan difícil, y en especial a Rigo por estar al tanto de mis responsabilidades en todo momento.

A mis tíos y primos, por tener seguridad en que lograría este éxito, en especial a la memoria de mi tío Jesús.

A todos mis amigos, especialmente, Anita, Yasnalla, Tayche, Ricardo, Pla, Yasmín y Valía, gracias por estar a mi lado, darme su apoyo y amistad cuando más los necesitaba.

A mi compañera de tesis, que es ante todo mi amiga, gracias por tanta paciencia, ayuda, amistad y consejos que me ayudaron a salir adelante cada vez que lo necesitaba.

A mami Ame, una gran amiga de mi mamá, por estar como una segunda madre y aconsejarme en los momentos que me hicieron falta.

A mi familia en general por ser tan unida y maravillosa.

A todos los profesores que de una forma u otra contribuyeron a que este sueño se hiciera realidad y por brindarme todos sus conocimientos.

Tailys

AGRADECIMIENTOS

Se agradece en general:

A Delfy y Héctor, por su paciencia, comprensión, ayuda y largas horas de empeño para que este trabajo se hiciera realidad.

A nuestra tutora Yamilis, por brindarnos su apoyo y dedicarnos tanto tiempo y esfuerzo.

A Liuris, Leevan, Fernando Peón, Amari, Guille y Gretter por haber colaborado con la realización de esta investigación.

A nuestra Revolución y a nuestro comandante en Jefe, por ayudarnos hacer realidad nuestro sueño.

DEDICATORIA

Yanisleidy

A mis padres, hermano, abuelas, a la memoria de mi abuelo y a mi familia en general, por ser mi gran motivo de inspiración y por tener siempre plena confianza en mí.

Tailys

A mis padres, abuelos y a toda mi familia, por ser mi luz y guía.

A la memoria de mi abuela Lala y mi tío Jesús, que siempre los llevo en mi corazón.

Dedicamos el presente trabajo, a una persona muy especial, que sino hubiese sido por su magnífica idea, de crear la Universidad de las Ciencias Informáticas, muchos de nuestro sueños no se hubiesen hecho realidad. A ti Fidel por siempre confiar en nosotros, tu Tropa de Futuro.

A Fidel Castro Ruz

RESUMEN

En el mundo del desarrollo de software, realizar una buena Gestión de Riesgo en todo el Proceso de Desarrollo implica lograr un producto con buena Calidad a un menor costo y que se entregue en tiempo.

En esta investigación esta dividida en tres Capítulos.

El primero es un estudio del arte donde se ofrecen precisiones conceptuales sobre, Procesos de Desarrollo de Software, también algunos elementos de Calidad, pero este estudio es guiado principalmente a la Gestión de Riesgo la cual incluye: su definición, las estrategias que se llevan a cabo para la realización de la misma, los tipos de riesgos existentes, sus atributos y una descripción de los pasos que se realizan en la Gestión de Riesgo.

En segundo lugar esta enmarcado en realizar una Propuesta de una serie de pasos para Gestionar Riesgo en los proyectos productivos de la Universidad de las Ciencias Informáticas, para lograr este objetivo se realiza una caracterización de la UCI mediante una dos encuestas, donde la línea fundamental es la Gestión de Riesgo y el Proceso de Desarrollo de Software, luego de haber dado este paso se arriba a varias conclusiones, una de ellas que las Metodología que mas se utilizaba en el centro es el *Rational Unified Process*, por lo se decidió llevar a cabo un estudio sobre esta Metodología, para de ahí realizar la propuesta antes mencionada.

En tercer lugar es realizar una valoración de esa propuesta, la cual pasara por manos de un grupo de expertos.

PALABRAS CLAVE

Proceso de Desarrollo de Software, Gestión de Riesgo

INDICE

AGRADECIMIENTOS	I
AGRADECIMIENTOS	II
AGRADECIMIENTOS	III
DEDICATORIA	IV
RESUMEN	V
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1. ¿Qué es Calidad Total?	5
1.2. ¿Qué se define como Calidad del Software?	6
1.3. Proceso de Desarrollo de Software	8
1.3.1. ¿Qué se define como Proceso de Desarrollo de Software?.....	8
1.3.2. Actividades del Proceso de Desarrollo de Software.....	11
1.3.3. Modelos del proceso de desarrollo del software	12
1.3.4. ¿Cuál es el modelo de proceso más adecuado?	18
1.3.5. Metodologías para el Proceso de Desarrollo del Software.....	19
1.4. Gestión de Riesgo	29
1.4.1. ¿Qué es Riesgo?	29
1.4.2. Estrategias Reactiva y Pro - activa	30
1.4.3. Tipos de Riesgos.....	31
1.4.4. Métodos generales de resolución de Riesgos	33
1.5. ¿Qué es Gestión?	34
1.5.1. ¿Qué es la Gestión de Riesgo?.....	34
1.5.2. Historia de la Gestión de Riesgo.....	35
1.5.3. Modelos que esquematizan la Gestión de Riesgo	36

1.5.4.	Explicación detallada de cada uno de los Pasos.....	39
1.5.5.	Resumen de los pasos de la Gestión de Riesgo	51
1.5.6.	Herramientas usadas para la Gestión de Riesgo	55
1.6.	Conclusiones.....	56
 CAPÍTULO 2: PROCESO DE GESTIÓN DE RIESGO EN LOS PROYECTOS PRODUCTIVOS		57
2.1.	Caracterizar la Gestión de Riesgo y el Proceso de Desarrollo de Software en los Proyectos Productivos	57
2.1.1.	Caracterizar el Proceso de Desarrollo de Software	57
2.1.2.	Caracterizar la Gestión de Riesgos	58
2.1.3.	Caracterización de las encuestas	58
2.2.	La Administración de Riesgos según el <i>Rational Unified Process</i>	58
2.2.1.	Conceptos fundamentales que brinda RUP sobre La Administración de Riesgos	61
2.2.2.	Pasos que define para la Gestión de Riesgos	62
2.3.	Proceso para la Gestión de Riesgo.....	65
2.3.1.	Flujo de Trabajo: Identificación	67
2.3.1.1.	Asignar un equipo para la Gestión de Riesgos	67
2.3.1.2.	Identificar los riesgos	67
2.3.1.3.	Determinar la lista de riesgos potenciales.....	68
2.3.1.4.	Aplicar las listas de comprobación de riesgos.....	68
2.3.1.5.	Determinar las herramientas para la gestión de riesgos	68
2.3.2.	Flujo de Trabajo: Análisis	69
2.3.2.1.	Analizar los riesgos según la probabilidad	70
2.3.2.2.	Valorar el Impacto y establecer una categoría	70
2.3.2.3.	Calcular la Exposición de riesgo (marco de tiempo).....	71
2.3.2.4.	Realizar un ordenamiento según la Probabilidad y el Impacto	71
2.3.3	Flujo de Trabajo: Priorización	71
2.3.3.1	Determinar el nivel de los riesgos a monitorear	72
2.3.3.2.	Seleccionar los 10 riesgos de mayor relevancia	73
2.3.3.3.	Dar prioridad a cada riesgo.....	73

2.3.4. Flujo de Trabajo: Planificación	74
2.3.4.1. Seleccionar estrategias para administrar los 10 riesgos	74
2.3.4.2. Determinar un cronograma.....	74
2.3.4.3. Elaborar el Plan de Contingencia	75
2.3.5. Flujo de Trabajo: Resolución y/o Mitigación	75
2.3.5.1. Implementar las estrategias	76
2.3.5.2. Poner en práctica el Plan de Contingencia.....	76
2.3.6. Flujo de Trabajo: Monitoreo	77
2.3.6.1. Realizar un Monitoreo	77
2.3.6.2. Realizar un control de la lista de riesgo durante la iteración	78
2.3.6.3. Revisar la lista de riesgo al final de la iteración	78
2.3.6.4. Realizar un control periódico	78
2.4. Capacitación del Personal	78
2.4.1 Sistema de contenido de Proceso de Desarrollo de Software	79
2.4.2. Sistema de contenido de Gestión de Riesgo	79
2.5. Conclusiones	79
CAPÍTULO 3: EVALUACIÓN DE LA PROPUESTA	80
3.1. Proceso de selección de los expertos	80
3.1.1. Determinar la cantidad de expertos	80
3.1.2. Confeccionar el listado de expertos.....	81
3.1.3. Confirmar la participación de los expertos	82
3.2. Elaboración del cuestionario	82
3.3. Resultados de la evaluación	83
3.4. Conclusiones	84
CONCLUSIONES	85
RECOMENDACIONES	86
BIBLIOGRAFÍA	87

ANEXO I	93
ANEXO II	98
ANEXO III	100
ANEXO IV	102
ANEXO V	103
ANEXO VI	105
GLOSARIO	106

INTRODUCCIÓN

En la actualidad la producción de software es un pilar muy importante que ha llegado a involucrar a una cantidad considerable de naciones, organizaciones y profesionales de Informática y la Computación, empeñados en mantener o lograr posiciones prestigiosas en el mercado.

Cuba, país subdesarrollado, sometido a numerosas presiones externas que impiden su natural desarrollo tecnológico como por ejemplo la Biotecnología, Bioinformática, la Producción de Biofármacos, y que se originan en aquellos países que detentan el poder económico, tecnológico y militar, ha elaborado una política inteligente y justa de acceso a las TIC. El objetivo principal de esta política está en el uso equitativo, racional, igualitario y educativo, de estas tecnologías, teniendo como fin ultimo el aumento espiritual y material del hombre (GIL, 2006).

Este proyecto cubano de profundo contenido social se consolida día a día con el incremento de diversos programas de la Revolución, relacionados con el acceso a las TIC en el país, la formación del capital humano calificado para el uso óptimo de éstas y la formación de la sociedad. Se debe destacar entre ellos, la introducción de la computación desde los primeros grados de la enseñanza primaria, incluso desde el grado preescolar, las escuelas Secundaria Básica en todo el país, los Jóvenes Club de Computación y Electrónica que suma alrededor de 600 en todo el territorio nacional, los Institutos Politécnicos de Informática (IPI), la formación de Ingenieros Informáticos en 13 universidades de todo el país y Licenciados en Ciencia de la Computación en otras tres, así como la creación de la Universidad de las Ciencias Informáticas (UCI), que actualmente se encuentra cursando su quinto año académico con una matricula de 10 000 estudiantes de todo el país, además de contar con tres Facultades Regionales en tres provincias del territorio (GIL, 2006).

En esta nueva casa de estudio, que une formación con producción, se desarrolla decenas de proyectos productivos, algunos de ellos dedicados a la exportación y otros a lograr la Informatización de la Sociedad. Su línea de producción aboga porque cada uno de sus proyectos tenga como prototipo lo planteado por el CMM (*Capability Maturity Model*), el cual define una serie de categorías donde está incluida la Gestión de Proyectos y esta a su vez tiene una amplia área de proceso que dentro de ella contiene la Gestión de Riesgos.

La Gestión de Riesgos es la práctica compuesta por procesos, métodos y herramientas que posibilitan la Gestión de los mismos en un proyecto, desde su inicio y a lo largo de todo su ciclo de vida. Esta actividad procura formalizar conocimiento orientado a la identificación, minimización y “evitación” de los riesgos más importantes, desastres, re trabajo, y sobre – trabajo en proyectos de desarrollo de software, mediante la generación de principios y buenas prácticas de aplicación realista (ROPPONEN y LYYTINEN, 2000; ROSENBERG *et al.*, 1999; SEI, 2004).

Hasta el momento se ha propuesto y utilizado diferentes enfoques de Gestión de Riesgos, desde que Boehm (1988) atrajo a la Comunidad Internacional de Ingeniería de Software hacia esta práctica o actividad. Sin embargo, es evidente que pocas organizaciones utilizan todavía de una forma explícita y sistemática, métodos específicos para Gestionar los riesgos en sus proyectos (SÁNCHEZ, 1997).

Según (SÁNCHEZ, 1997) plantea que las tareas principales de la Gestión de Riesgos son: involucrar al equipo de desarrollo de proyecto y a los afectados/implicados, en el proceso de Identificación de Riesgos; comunicar los mismos a todos los niveles de la organización y monitorear constantemente los factores que propician la materialización de los riesgos y la efectividad de las acciones, encaminadas a su prevención y/o minimización.

En tal sentido, todo proyecto informático debe proponerse, minimizar el impacto de los riesgos y obtener un producto con Calidad, teniendo en cuenta que ésta no solo se relaciona con el producto sino también con el proceso. Así se contribuye a un mejoramiento continuo de la calidad, para ofrecerle al cliente, productos y servicios que cumplan con sus necesidades y exigencias.

La UCI es una entidad que puede favorecer al crecimiento de la economía, debido a que la mayoría de sus proyectos presentan convenios con otros países. Viéndose en la necesidad entonces, de capacitar al personal vinculado a los proyectos, donde se logre una formación con sólidos conocimientos teóricos y amplias habilidades prácticas, basada en dos temáticas principales: Proceso de Desarrollo de Software y Gestión de Riesgo.

Según estudios realizados acerca de cómo se administran los riesgos en la Universidad, se llega a la conclusión de que existen situaciones adversas que se identifican como riesgos, los cuales atentan contra

la calidad del producto y con el desarrollo del proyecto, ocasionando: atrasos en los cronogramas, en la entrega del producto y dificultades con la culminación exitosa del software.

Además, en la universidad no existe una vasta cultura referente a este tema, ni un procedimiento para administrar o prever los riesgos durante todo el ciclo de vida del software. Por tanto la UCI se ve en la necesidad de asumir un procedimiento para llevar a cabo la Gestión de Riesgo en cada uno de sus proyectos productivos e ir buscando otras alternativas para garantizar la formación del personal vinculado a la producción.

El presente trabajo tiene el fin de dar respuesta a la siguiente pregunta:

¿Cómo realizar la Gestión de Riesgo durante todo el Proceso de Desarrollo de Software en los proyectos productivos de la UCI, a la vez que se prepara al personal que realizará dicha gestión?

Por ende, esta pregunta indica que el **objeto de estudio** de esta investigación lo constituye: el Proceso de Desarrollo de Software en los proyectos productivos UCI y su **campo de acción** lo conforma: la Gestión de Riesgo dentro de los proyectos productivos de la UCI.

Objetivos

Objetivo General

Proponer un procedimiento para la Gestión de Riesgos en los proyectos productivos de la UCI, garantizando la capacitación del personal que lo ejecutará.

Objetivos Específicos

- Obtener un proceso de Gestión de Riesgo adaptable a los proyectos productivos de la UCI.
- Diseñar el sistema de contenido de un curso de capacitación para los estudiantes vinculados a la producción.

Para dar cumplimiento a los objetivos esbozados anteriormente, se hace necesario que se cumplan las siguientes tareas:

- Realizar un estado del arte sobre la Calidad Total, Calidad de Software y el Proceso de Desarrollo de Software y la Gestión de Riesgo.
- Caracterizar el Proceso de Desarrollo de Software y la Gestión de Riesgo.
- Definir un proceso para la Gestión de Riesgo durante todo el Proceso de Desarrollo de Software en los proyectos productivos.

- Diseñar el sistema de contenido de un curso de capacitación para formar al personal vinculado a proyectos, basado en dos temáticas fundamentales: Proceso de Desarrollo de Software y Gestión de Riesgo.
- Validar el procedimiento propuesto, aplicando el criterio de un Grupo de Expertos.

Este trabajo cuenta con la Introducción, tres capítulos, Conclusiones y Recomendaciones.

Capítulo I, se hace un estado del arte sobre los temas de Proceso de Desarrollo de Software y Gestión de Riesgo, así como referencias a conceptos que son utilizados a lo largo del trabajo.

Capítulo II, se describe de forma sencilla y explícita el proceso que se propone y que es el objetivo principal en este trabajo. La descripción se realiza de manera gráfica y textual. Además de brindar la propuesta del curso de capacitación para los estudiantes de los proyectos productivos.

Capítulo III, se presenta la evaluación del proceso teniendo en cuenta los resultados que arrojaron las respuestas del criterio del Grupo de Expertos.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se hará un estudio de diferentes terminologías como: Calidad Total, Calidad de Software, Gestión de Proyecto, Proceso de Desarrollo de Software y Gestión de Riesgo, esta última constituye el eje principal en este Trabajo de Diploma. Esta investigación dará una idea de cómo se manejan dichas terminologías a nivel mundial.

El Proceso de Desarrollo de Software incluye el estudio de las actividades, los modelos y metodologías que en él se utilizan, como por ejemplo: RUP (*Rational Unified Process*), MSF (*Microsoft Solution Framework*) y XP (*Extreme Programming*).

En la Gestión de Riesgo se abordarán diferentes temas, lo primero será conocer y profundizar en los diferentes conceptos que plantean personalidades e Instituciones sobre Riesgo y Gestión de Riesgo. Luego se entrará a conocer más detalladamente los elementos fundamentales que contiene la Gestión de Riesgo como son: las Estrategias que se deben seguir para evitar que los riesgos puedan afectar al proyecto, los Tipos de Riesgos más comunes que se dan en los proyectos, los Atributos de cada riesgo y finalmente se analizarán los diferentes Modelos que se siguen para realizar una buena Gestión de Riesgo y dentro de esos modelos, se detallarán los pasos con mayor profundidad.

1.1. ¿Qué es Calidad Total?

La Calidad Total es una filosofía empresarial nacida en Japón, que se produce como consecuencia de la transformación de la industria japonesa tras la Segunda Guerra Mundial.

Importantes definiciones han sido citadas referentes a este término. Entre ellas, Plaza (2002), citado por (FIGUEROA, 2004) plantea que: “La Calidad Total es una estrategia de gestión integral de la empresa...”.

Por otra parte, la Calidad Total puede entenderse tal y como se sugiere en el Modelo EFQM (*European Foundation for Quality Model*), como una estrategia de gestión de toda la empresa, a través de la cual se satisfacen las necesidades y expectativas de los clientes, de los empleados, de los accionistas y de la sociedad en general; por medio de la utilización eficiente de todos los recursos de que dispone: personas, materiales, tecnologías, sistemas productivos, entre otras (ELCHE, 2001).

Según Ruiz-Canela (2004), este término implica también, una filosofía de trabajo y un camino hacia la excelencia. Es un paradigma o modelo de gestión, que persigue centrar la estrategia de la empresa en dar al cliente lo que necesita y cuando lo necesita, con un precio competitivo y de la manera más eficaz posible. Se despliega a partir de una decisión de la dirección de la organización, que deberá conseguir un compromiso de todas las personas. Esto se aprende mediante la formación, y se construye y se gestiona a través de su planificación, implantación, desarrollo, seguimiento y mejora (FIGUEROA, 2004).

La Calidad Total, surge cuando se comprende la importancia de la calidad para el éxito de los negocios, por ende su objetivo básico es la mejora de la gestión y los resultados de la organización en el más amplio sentido del término (FIGUEROA, 2004).

Las autoras de este trabajo, basadas en los conceptos antes expuestos definen la Calidad Total como: Una filosofía que pretende abarcar a toda la organización y a todas sus actividades. Es además, una estrategia de gestión de toda la empresa, donde se satisfacen las necesidades de los clientes, al menor costo y de la forma más eficaz posible a través de la utilización eficiente de todos los recursos que estén disponibles.

Para la mejor comprensión de este tema se citan, según (CONSULTORES, 2006), un conjunto de fundamentos básicos que implica la Calidad Total:

1. Orientar la organización hacia el cliente.
2. Ampliar el concepto de cliente.
3. Poseer liderazgo en costos.
4. Gestionar basándose en la prevención.
5. Potenciar el factor humano.
6. Mejora permanente.

1.2. ¿Qué se define como Calidad del Software?

En el mundo de la informática, abundan muchas ideas de lo que significa la Calidad del Software, debido al gran auge que ha cobrado a nivel global.

CAPITULO I. FUNDAMENTACION TEORICA

El profesor Roger S. Pressman (2005) uno de los grandes investigadores de la Ingeniería de Software define la Calidad de Software como: “La concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.” (PRESSMAN, 2005).

Según la IEEE (Std. 610-1990): La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario. Citado por (DIRECCIÓN DE LA CARRERA DE INFORMÁTICA DE LA UTEZ, 2005).

La norma internacional ISO (8402) la define también como: El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas (BUADES, 2002).

Teniendo en cuenta los conceptos antes expuestos, las autoras de esta investigación definen la Calidad de Software como Pressman, pero considerando el factor tiempo. El cual es muy importante a la hora de desarrollar un software, según los compromisos de entrega.

Cuando se habla de calidad del software, pueden distinguirse dos áreas principales: Calidad del Producto y Calidad del Proceso de Desarrollo (ORANTES y CASTILLO, 2007).

Para la evaluación de la Calidad del Producto existen modelos y estándares que dictan procedimientos, actividades, características y atributos a tomar en cuenta, ya que es importante ofrecer productos de un alto grado de calidad y así, enfrentarse a la fuertemente creciente competitividad que existe en esa rama, no solamente a nivel local, sino incluso internacional.

Al evaluar la Calidad de los Procesos de Desarrollo de Software, se pretende determinar su rendimiento para corregir problemas o para mejorarlos, con el propósito de disminuir costos de desarrollo, aumentar la productividad y por consiguiente, elevar la calidad de los productos usando estándares internacionales considerados importantes para la evaluación de los procesos de desarrollo de software.

1.3. Proceso de Desarrollo de Software

1.3.1. ¿Qué se define como Proceso de Desarrollo de Software?

Una terminología importante a definir en este capítulo es, Proceso de Desarrollo de Software o Ciclo de Vida del Software:

Jacobson (1998) define: Proceso de Desarrollo de Software es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo. Concretamente define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo (ZAVALA, 2000).

Por otra parte, la IEEE lo define como: Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software (MEDINA, 2006).

Según la ISO (12207), Proceso de Desarrollo es: Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso (MEDINA, 2006).

Roger. S. Pressman, (PRESSMAN, 2005), en su libro: Un Enfoque Práctico, define Proceso de Desarrollo de Software, como: Un marco común del proceso, donde se define un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad, que constituye un número de conjuntos de tareas, donde cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto y finalmente las actividades de protección, tales como garantía de calidad del software, gestión de configuración del software y medición, que abarcan el modelo del proceso. Además, estas actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso. Ver Figura 1.

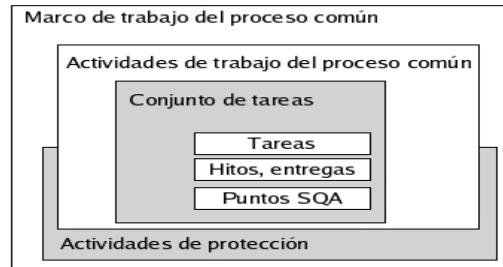


Figura 1 Elementos Proceso de Desarrollo de Software

Considerando los conceptos antes expuesto, las autoras de esta investigación definen Proceso de Desarrollo de Software o Ciclo de Vida del Software como:

Un marco común del proceso, donde se define un pequeño número de actividades del marco de trabajo, que constituyen a su vez, un conjunto de tareas, donde cada uno es una colección de hitos, puntos de garantía de calidad y tareas de ingeniería de software. Donde las necesidades del usuario son traducidas en requerimientos, estos transformados en diseño, el diseño es implementado en código, y finalmente el software es probado, documentado y certificado. Además de establecer, las actividades de protección. Concretamente, Proceso de Desarrollo de Software define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo.

Las relaciones entre los elementos del Proceso de Desarrollo de Software que permiten responder **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo, se exponen en la Figura 2, y así las interrogantes se responden de la siguiente forma:

- **Quién:** Las Personas participantes en el proyecto de desarrollo desempeñando uno o más Roles específicos.
- **Qué:** Un Artefacto es una pieza de información que es producida, modificada o usada por el proceso, define un área de responsabilidad para un rol y está sujeta a control de versiones. Un artefacto puede ser un modelo, un elemento de modelo o un documento. Es producido por un Rol en una de sus actividades. Los artefactos se especifican utilizando notaciones específicas. Las herramientas apoyan la elaboración de artefactos soportando ciertas notaciones.
- **Cómo y Cuándo:** Las Actividades son una serie de pasos que lleva a cabo un Rol durante el proceso de desarrollo. El avance del proyecto está controlado mediante hitos que establecen un determinado estado de terminación de ciertos Artefactos.

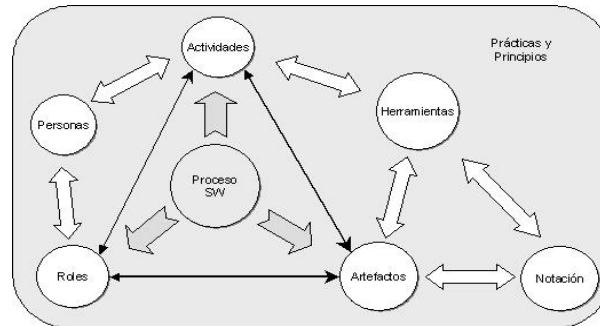


Figura 2 Relación entre elementos del Proceso de Desarrollo de Software

Un Proceso de Desarrollo de Software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente (VALENCIA, 2003). Dicho proceso, a nivel global se muestra en la Figura 3.



Figura 3 Proceso de Desarrollo de Software

Aunque un proyecto de desarrollo de software es comparable en varios aspectos a cualquier otro proyecto de ingeniería, en el desarrollo del software existen una serie de desafíos adicionales, relativos a la naturaleza del producto obtenido y particularidades que influyen en su proceso de construcción, como por ejemplo:

- Un producto software en sí es complejo y es prácticamente imposible conseguir un 100% de confiabilidad de un programa por pequeño que sea. Existe una inmensa combinación de factores que impiden una verificación exhaustiva de todas las posibles situaciones de ejecución que se puedan presentar (entradas, valores de variables, datos almacenados, software del sistema, otras aplicaciones que intervienen, el hardware sobre el cual se ejecuta, entre otras).
- Un producto software es intangible y por lo general muy abstracto, esto dificulta la definición del producto y sus requisitos, sobre todo cuando no se tiene precedentes en productos software similar. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado el producto sino también durante el proceso de desarrollo.

- Además, de las dos anteriores, siempre se puede señalar la inmadurez de la ingeniería del software como disciplina, justificada por su corta vida comparada con otras disciplinas de la ingeniería.
- Mayor énfasis en la abstracción, en el modelado, en la representación, organización de la información y en la administración del cambio.

1.3.2. Actividades del Proceso de Desarrollo de Software

Un proceso de desarrollo de software no es único y no existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. A pesar de esta variedad, existen un conjunto de actividades fundamentales que se encuentran presentes en todos ellos (VALENCIA, 2003), estas son:

- **Análisis (Negocio y Requerimiento):** Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.
- **Diseño:** Se diseña el software.
- **Implementación:** Se construye el software de acuerdo a la especificación.
- **Validación, verificación y prueba:** El software debe validarse, verificarse y probarse para asegurar que cumpla con lo que quiere el cliente.
- **Mantenimiento:** El software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, Pressman (1998) menciona otro conjunto de actividades en este caso las actividades protectoras. Ellas se señalan a continuación:

- Seguimiento y control de proyecto de software.
- Revisiones técnicas formales.
- Garantía de calidad del software.
- Gestión de configuración del software.
- Preparación y producción de documentos.
- Gestión de reutilización.
- Mediciones.
- Gestión de riesgos.
- Auditorias.

Estas actividades protectoras, se aplican a lo largo de todo el proceso del software, y son independientes de las otras actividades.

1.3.3. Modelos del proceso de desarrollo del software

Un modelo del proceso de desarrollo o ciclo de vida de software define el estado de las fases a través de las cuales se mueve el proyecto, es una visión de las actividades que ocurren durante el desarrollo de software y determina el orden de las etapas involucradas en cada una de las actividades. Además, ayuda administrar y monitorear el progreso del software suministrando guías para los ingenieros de software con el fin de ordenar las diversas actividades técnicas.

Modelos que se van a exponer a continuación:

- Codificar y corregir
- Modelo en cascada
- Desarrollo evolutivo
- Desarrollo basado en reutilización
- Desarrollo iterativo e incremental
- Desarrollo en espiral

Codificar y corregir.

Este es el modelo básico utilizado en los inicios del desarrollo de software. Contiene dos pasos:

- Escribir código.
- Corregir problemas en el código.

Se trata de primero implementar algo de código y luego pensar acerca de requisitos, diseño, validación, y mantenimiento.

Este modelo tiene tres problemas principales:

- Después de un número de correcciones, el código puede tener una muy mala estructura, hace que los arreglos sean muy costosos.
- Frecuentemente, aún el software bien diseñado, no se ajusta a las necesidades del usuario, por lo que es rechazado o su reconstrucción es muy cara.
- El código es difícil de reparar por su pobre preparación para probar y modificar.

Modelo en cascada

Este es el más básico de todos los modelos y sirve como base de construcción para los restantes modelos del ciclo de vida.

En el modelo cascada, el desarrollo de software se realiza a través de una secuencia simple de fases, donde cada fase tiene un conjunto de metas bien definidas. Además, toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, representándolas como fases separadas del proceso.

El Modelo en Cascada consta de las siguientes fases (VALENCIA, 2003):

- Definición de los requisitos: Los servicios, restricciones y objetivos son establecidos con los usuarios del sistema. Se busca hacer esta definición en detalle.
- Diseño: Se particiona el sistema en sistemas de software o hardware. Se establece la arquitectura total del sistema. Se identifican y describen las abstracciones y relaciones de los componentes del sistema.
- Implementación y pruebas unitarias: Se construyen los módulos y unidades de software. Se realizan pruebas de cada unidad.
- Integración y pruebas del sistema: Se integran todas las unidades. Se prueban en conjunto. Se entrega el conjunto probado al cliente.
- Operación y mantenimiento: Generalmente es la fase más larga. El sistema es puesto en marcha y se realiza la corrección de errores descubiertos. Se realizan mejoras de implementación. Se identifican nuevos requisitos.

La interacción entre las fases se puede observar en la Figura 4. Las flechas muestran el flujo de información entre las fases, las de avance muestran el flujo normal y las que van en sentido contrario, la retroalimentación.

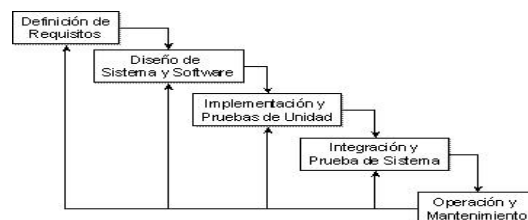


Figura 4 Modelo en Cascada

Cada fase tiene como resultado documentos que deben ser aprobados por el usuario. Además, una fase no comienza hasta que termine la fase anterior y generalmente se incluye la corrección de los problemas encontrados en fases previas.

En la práctica, este modelo no es lineal, e involucra varias iteraciones e interacción entre las distintas fases de desarrollo (VALENCIA, 2003). Algunos problemas que se observan en el modelo de cascada son:

- Las iteraciones son costosas e implican rehacer trabajo debido a la producción y aprobación de documentos.
- Los problemas se dejan para su posterior resolución, lo que lleva a que estos sean ignorados o corregidos de una forma poco elegante.
- Existe una alta probabilidad de que el software no cumpla con los requisitos del usuario por el largo tiempo de entrega del producto.
- Es inflexible a la hora de evolucionar para incorporar nuevos requisitos. Es difícil responder a cambios en los requisitos.

Debido a los problemas que presenta el Modelo en Cascada, se sugiere que sólo debe usarse si se entienden a plenitud los requisitos, aunque se sigue utilizando como parte de proyectos grandes.

Desarrollo evolutivo

El propósito de este modelo es lograr el desarrollo de una implantación del sistema inicial, para exponérsela a los usuarios e ir la refinando a través de las versiones hasta lograr el desarrollo del sistema adecuado para ellos.

En la Figura 5 se observa como las actividades concurrentes, que no son más que, la especificación, el desarrollo y la validación, se realizan durante el desarrollo de las versiones hasta llegar al producto final.

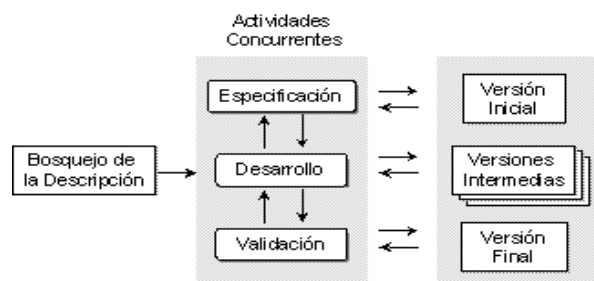


Figura 5 Modelo de desarrollo evolutivo

Existen dos tipos de desarrollo evolutivo, según (VALENCIA, 2003):

- Desarrollo Exploratorio: El objetivo de este enfoque es explorar con el usuario los requisitos hasta llegar a un sistema final, comenzando con las partes que se tiene más claras. El sistema evoluciona a medida que se añaden nuevas características propuestas por el usuario.
- Enfoque utilizando prototipos: El objetivo es entender los requisitos del usuario y trabajar para mejorar la calidad de los requisitos. A diferencia del desarrollo exploratorio, aquí se comienza por definir los requisitos que no están claros para el usuario y se utiliza un prototipo para experimentar con ellos. El prototipo ayuda a terminar de definir estos requisitos.

Son varias las ventajas que aporta este modelo, entre ellas se encuentran:

- Se obtiene una rápida realimentación del usuario, ya que las actividades de especificación, desarrollo y pruebas se ejecutan en cada iteración.
- Los usuarios y desarrolladores logran un mejor entendimiento del sistema, reflejando una mejora en la calidad del software.
- Cumple con las necesidades inmediatas del usuario.

A pesar de poseer puntos favorables, el desarrollo de software en forma evolutiva requiere un especial cuidado en la manipulación de documentos y programas desarrollados para las distintas versiones del software. Cada paso debe ser registrado, la documentación debe ser recuperada con facilidad y los cambios deben ser efectuados de una manera controlada.

Este modelo se propone aplicarlos en proyectos pequeños, menos de 100.000 líneas de código, o proyectos medianos, hasta 500.000 líneas de código, con poco tiempo de desarrollo, logrando así que sea mas efectivo.

Desarrollo basado en reutilización

Este es un modelo orientado a la reutilización, como su nombre lo indica. Consta de cuatro fases: Análisis de componentes, Modificación de requisitos, Diseño del sistema con reutilización y el Desarrollo e integración, las cuales se ilustran en la Figura 6 y se describen a continuación.

- Análisis de componentes: Se determina qué componentes pueden ser utilizados para el sistema en cuestión, realizando casi siempre ajustes para adecuarlos.

- **Modificación de requisitos:** Se adaptan los requisitos para coincidir con los componentes de la etapa anterior, en caso de no poder realizar modificaciones en los requisitos, hay que buscar los componentes más adecuados (Fase 1).
- **Diseño del sistema con reutilización:** Se diseña o reutiliza el marco de trabajo para el sistema, teniendo en cuenta los componentes localizados en la Fase 2, para determinar este marco.
- **Desarrollo e integración:** Se desarrolla el software que no puede comprarse, se integran los componentes y subsistemas, siendo la integración parte del desarrollo, en lugar de ser una actividad separada.

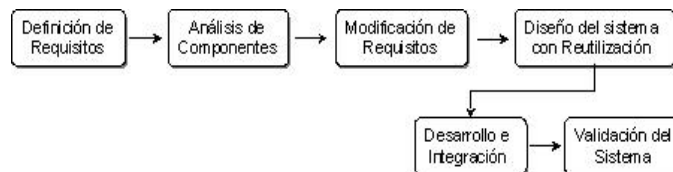


Figura 6 Desarrollo basado en reutilización de componentes

El modelo de Desarrollo de Reutilización, es capaz de disminuir el costo, el esfuerzo y los riesgos durante el desarrollo, además de reducir el tiempo de entrega.

Como todo modelo tiene sus desventajas, debido a que los “compromisos” en los requisitos son inevitables, por lo cual puede que el software no cumpla las expectativas del cliente, y las actualizaciones de los componentes adquiridos no están en manos de los desarrolladores del sistema.

A continuación se expondrán dos enfoques híbridos, especialmente diseñados para el soporte de las iteraciones:

- Desarrollo iterativo e incremental.
- Desarrollo en Espiral.

Desarrollo iterativo e incremental

Es una forma de reducir el rehacer del trabajo durante el proceso de desarrollo y ofrecer oportunidad para retrasar las decisiones hasta alcanzar experiencia en el sistema Figura 7.

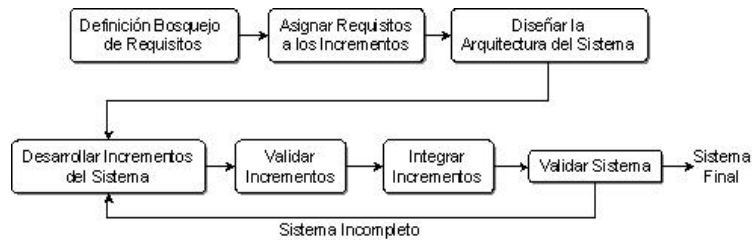


Figura 7 Modelo de desarrollo iterativo e incremental

Este modelo es una combinación del Modelo de Cascada y Modelo Evolutivo, que se pueden utilizar durante el desarrollo de cada incremento, en dependencia del conocimiento que se tenga sobre los requisitos a implantar. En caso que tenga un buen conocimiento, se puede optar por cascada, si es dudoso, evolutivo (VALENCIA, 2003).

Entre las ventajas que presenta el Modelo Incremental se encuentran:

- Los clientes no tiene que esperar al final para utilizar el sistema lo pueden hacer desde el primer incremento.
- Los clientes pueden aclarar los requisitos que no tengan claros conforme ven las entregas del sistema.
- Se disminuye el riesgo de fracaso de todo el proyecto, al ser posible distribuir en cada incremento.
- Las partes más importantes del sistema son entregadas primero.

No obstante a las grandes ventajas que ofrece este modelo, tiene sus desventajas, dadas porque:

- Cada incremento debe ser pequeño para limitar el riesgo (menos de 20.000 líneas).
- Cada incremento debe aumentar la funcionalidad.
- Es difícil establecer las correspondencias de los requisitos contra los incrementos.
- Es difícil detectar las unidades o servicios genéricos para todo el sistema.

Desarrollo en espiral

El modelo de Desarrollo en Espiral es actualmente uno de los más conocidos y fue propuesto por Boehm. El ciclo de desarrollo se representa como una espiral (Figura 8), en lugar de una serie de actividades sucesivas con retrospectiva de una actividad a otra.

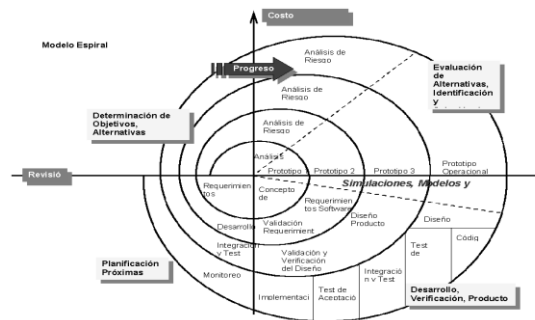


Figura 8 Modelo de desarrollo en Espiral

Cada ciclo de desarrollo se divide en cuatro fases:

- Definición de objetivos: Se definen los objetivos y las restricciones del proceso y del producto. Se realiza un diseño detallado del plan administrativo, se identifican los riesgos y se elaboran estrategias alternativas dependiendo de estos.
- Evaluación y reducción de riesgos: Se realiza un análisis detallado de cada riesgo identificado, pueden desarrollarse prototipos para disminuir el riesgo de requisitos dudosos y se llevan a cabo los pasos para reducir los riesgos.
- Desarrollo y validación: Se escoge el modelo de desarrollo después de la evaluación del riesgo. El modelo que se utilizará (cascada, evolutivo, entre otros) depende del riesgo identificado para esa fase.
- Planificación: Se determina si continuar con otro ciclo y se planea la siguiente fase del proyecto.

Este modelo a diferencia de los otros, toma en consideración explícitamente el riesgo, actividad importante en la administración del proyecto.

1.3.4. ¿Cuál es el modelo de proceso más adecuado?

Cada proyecto de software requiere una forma particular de abordar el problema. Las propuestas comerciales y académicas actuales promueven procesos iterativos, donde en cada iteración puede utilizarse uno u otro modelo de proceso, considerando un conjunto de criterios.

En la Tabla 1 se expone un cuadro comparativo de acuerdo con algunos criterios básicos, para la selección de un modelo de proceso, la medida utilizada indica el nivel de efectividad del modelo de proceso de acuerdo al criterio.

Tabla 1 Comparación entre modelos de proceso de software

Modelo de Proceso/ Criterio	Desempeño si no se predefinen requisitos	Produce software fiable	Gestión de Riesgos	Permite cambios sobre la marcha	Visibilidad del Progreso
Codificar y Corregir	Bajo	Bajo	Bajo	Alto	Medio
Desarrollo en Cascada	Bajo	Alto	Bajo	Bajo	Bajo
Desarrollo Evolutivo	Alto	Medio	Medio	Alto	Alto
Desarrollo Basado en Reutilización	Medio	Bajo a Alto	Bajo a Medio	Alto	Alto
Desarrollo Incremental	Bajo	Alto	Medio	Medio	Bajo
Desarrollo en Espiral	Alto	Alto	Alto	Medio	Medio

1.3.5. Metodologías para el Proceso de Desarrollo del Software

Metodología, es una forma de plantear un problema, de construir una solución, verificando siempre que sea la correcta. Es además, un proceso repetible, donde se definen los artefactos, roles y actividades involucradas, teniendo en cuenta cuatro componentes básicos: guías, técnicas, herramientas y plantillas.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtienen son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos (REYNOX, 2005), pues las metodologías aparecen por la necesidad de poner orden y guiar al proceso de construcción del software.

Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses, pero cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí si toma sentido el basarse en una metodología de desarrollo, y se empieza a buscar cual será la mas apropiada para el caso, ya que la elección de la metodología adecuada es mas importante que utilizar las mejores y más potentes herramientas (REYNOX, 2005).

Existen dos grandes grupos de metodologías a utilizar las cuales se explicarán a continuación, las Metodologías Tradicionales, dentro de esta: *RUP (Rational Unified Process o Proceso Unificado de Desarrollo)*, *MSF (Microsoft Solution Framework)*, entre otras y las *Metodologías Ágiles*, teniendo dentro de ellas, *XP (Extreme Programing)*, *Cristal*, *Scrum* y *Dsdm (Dynamic Systems Development Method)*.

1.3.5.1. Las Metodologías Tradicionales (no ágiles)

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible.

Son metodologías que no se adaptan adecuadamente a los cambios, por lo que no son métodos apropiados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. Están guiadas por una fuerte planificación durante todo el proceso de desarrollo, donde se realiza una intensa etapa de análisis y diseño antes de la construcción del sistema, pero cuando se produce un cambio, toda esta planificación se puede perder. Además, tiene un proceso poco flexible y más controlado, con numerosas políticas y normas.

Rational Unified Process (RUP)

La metodología RUP, llamada así por sus siglas en inglés *Rational Unified Process*, es guiado por casos de uso, es iterativo e incremental, centrado en la arquitectura y guiados por los riesgos. Además, se divide en 4 fases el desarrollo del software (Figura 9):

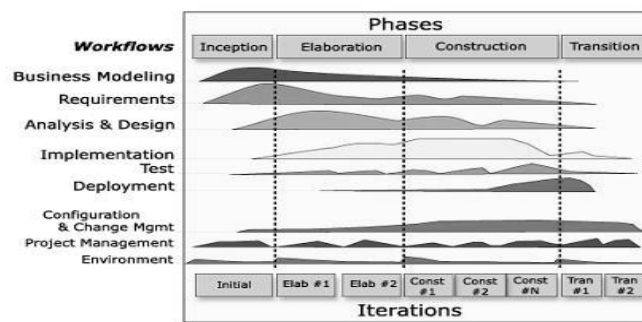


Figura 9 Metodología Rational Unified Process (RUP)

- **Inicio:** El Objetivo en esta etapa definir el alcance del proyecto, entender que se va a construir, es decir, determinar la visión del proyecto.

- **Elaboración:** En esta etapa el objetivo es construir una versión ejecutable de la arquitectura de la aplicación y entender cómo se va a construir.
- **Construcción:** En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial, construir una versión Beta.
- **Transición:** El objetivo poner a disposición la aplicación para los usuarios finales, es decir, llegar a obtener el release del proyecto.

Todas las fases antes mencionadas aportan un resultado y esto es reflejado en los cada uno de sus hitos (Figura 10).

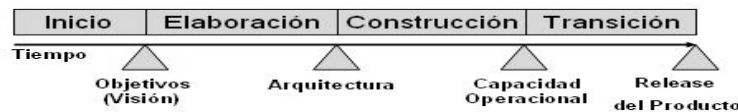


Figura 10 Hitos de cada una de las fases de RUP

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, y el ciclo de vida que se desarrolla por cada una de las iteraciones, es desarrollado por dos grupos (MENDOZA, 2004):

Disciplina de Desarrollo

- **Modelamiento del Negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- **Instalación o despliegue:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, entre otros.) para entregar el software a los usuarios finales.

Disciplina de Soporte

- Configuración y administración del cambio: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, entre otros.
- Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

RUP contiene una serie de elementos que son:

- Las actividades: procesos que se llegan a determinar en cada iteración.
- Los trabajadores: viene hacer las personas o antes involucrados en cada proceso.
- Los artefactos: estos pueden ser un documento, un modelo o un elemento de modelo.

Los elementos de RUP se visualizan en la figura 11.

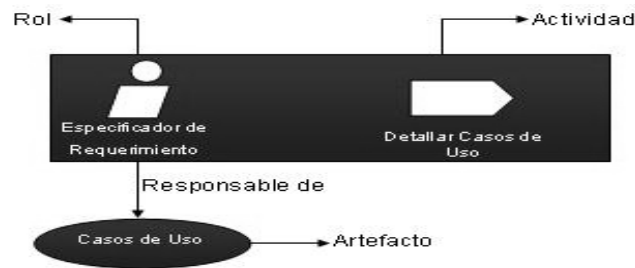


Figura 11 Elementos de RUP

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software, siendo adaptable para proyectos de largo plazo.

Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos, además de proporcionar principios y pautas para dar soluciones a empresas que diseñan y desarrollan, de manera que se asegure de que todos los elementos de un proyecto, tales como gente, procesos, y herramientas, puedan ser manejados con éxito.

MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

Concretamente MSF se compone de principios que están dados por (GATTACA, 2006):

- Promover comunicaciones abiertas
- Trabajar para una visión compartida
- Fortalecer los miembros del equipo
- Establecer responsabilidades claras y compartidas
- Permanecer ágil y esperar los cambios
- Invertir en calidad
- Aprender de todas las experiencias

Además, esta metodología está compuesta por modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto, ellos son (MENDOZA, 2004) :

- Modelo de Arquitectura del Proyecto
- Modelo de Equipo
- Modelo de Proceso
- Modelo de Gestión del Riesgo
- Modelo de Diseño de Proceso
- Modelo de Aplicación.

MSF combina los mejores principios del modelo en cascada y del modelo en espiral. Combina la claridad que planea el modelo en cascada y las ventajas de los puntos de transición del modelo en espiral.

Consta de cinco fases distintas (Figura 12):

- Visión y alcances

- Planificación
- Desarrollo
- Estabilización
- Implementación



Figura 12 Metodología *Microsoft Solution Framework (MSF)*

Visión y Alcances:

La fase de visión y alcances tiene como objetivo obtener una visión del proyecto compartida, comunicada, entendida y alineada con los objetivos del negocio. Además, identificar los beneficios, requerimientos funcionales, sus alcances y restricciones; y los riesgos inherentes al proyecto.

Esta fase trata también, la unificación del equipo detrás de una visión común. El equipo debe tener una visión clara de lo que quisiera lograr para el cliente y ser capaz de indicarlo en términos que motivarán a todo el equipo y al cliente. Se definen los líderes y responsables del proyecto.

Los entregables en esta fase son:

- Documento visión
- Documento detalle de la visión
- Documento Requerimientos Funcionales (Incluye *Script* de Pruebas)
- Documento Matriz de Riesgos
- Acta de aprobación de visión.

Planificación:

Esta fase tiene como objetivo llevar a cabo el diseño de la solución, obtener un cronograma de trabajo que cumpla con lo especificado en la fase de Visión dentro del presupuesto, tiempo y recursos acordados para

los diferentes entregables del proyecto. Este cronograma debe identificar puntos de control específicos que permitan generar entregas funcionales y cortas en el tiempo.

La planificación tiene como entregables:

- Documento de Cronograma
- Acta de aprobación de cronograma

Desarrollo:

El objetivo de la fase de Desarrollo es realizar la mayor parte de la construcción de los componentes (tanto documentación como código). Además, obtener de forma iterativa, de la fase de Planeación y de la Estabilización, versiones del producto entregable y medible que permitan de cara al cliente probar características nuevas de forma sucesiva. Esto incluye los ajustes de cronograma.

Los entregables en esta fase están dados por:

- Fuentes y ejecutables (según lo acordado)
- Documentos Manuales técnicos, de usuario y de instalación si es necesario
- Acta de finalización de Desarrollo

Estabilización:

La Estabilización tiene por objetivo lograr una versión final del producto probada, ajustada y aprobada en su totalidad.

Se conducen pruebas sobre la solución, enfatizando en el uso y operación bajo condiciones realistas. El equipo se enfoca en priorizar y resolver errores y preparar la solución para el lanzamiento.

Entregables:

- Documento Registro de pruebas
- Acta de aprobación de Versión Aprobada

Implantación:

Durante esta fase el equipo implanta la tecnología base y los componentes relacionados, estabiliza la instalación, traspasa el proyecto al personal soporte y operaciones, y obtiene la aprobación final del cliente.

Según lo antes expuesto se puede indicar que la metodología MSF se adapta a proyectos de cualquier dimensión, desde proyectos pequeños a proyectos largos y complejos, y de cualquier tecnología.

1.3.5.2. Las Metodologías Ágiles

En el proceso de buscar soluciones hoy en día para proyectos pequeños, con plazos reducidos y nuevas tecnologías, se encuentran las llamadas Metodologías Ágiles. Estas metodologías se centran en otros aspectos, como por ejemplo el factor humano o el producto de software, brindando mayor importancia a la colaboración con el cliente para que el mismo pueda cambiar los requisitos cuando desee y que el código funcione bien.

Además, los métodos ágiles ofrecen una comunicación en tiempo real, preferentemente cara a cara y sobre documentos escritos, donde involucre a todas las personas necesarias; programadores y clientes, probadores, diseñadores de interacción, escritores técnicos y gerentes, para garantizar la finalización del software (ORANTES, 2007).

Esta Metodología Ágil intenta ser lo más flexible posible, minimiza los riesgos en tiempos de desarrollo de software cortos, a través de iteraciones que duran de una a cuatro semanas y donde cada iteración es como un proyecto de software en miniatura, mostrando su efectividad donde se exiga reducir drásticamente el tiempo de desarrollo, pero manteniendo siempre una alta calidad (ORANTES, 2007).

A continuación se hará alusión a cuatro puntos que son los que se conocen como el “Manifiesto Ágil”, en el que se valoran las siguientes ideas (CANÓS *et al.*, 2006):

- Individuos e interacción frente a procesos y herramientas.
- Software que funciona frente a documentación exhaustiva.
- Colaboración del cliente frente a contratos.
- Responder al cambio frente a seguir el plan.

Los valores anteriores inspiran los doce principios del manifiesto. Siendo características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo (CANÓS *et al.*, 2006).

Los principios son:

- I. Satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- III. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- VII. El software que funciona es la medida principal de progreso.
- VIII. Promover un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- IX. Atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X. Simplicidad.
- XI. Mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizadas para proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer (REYNOX, 2005).

La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, pues constituye uno de los requisitos para llegar al éxito del proyecto, figura 13.

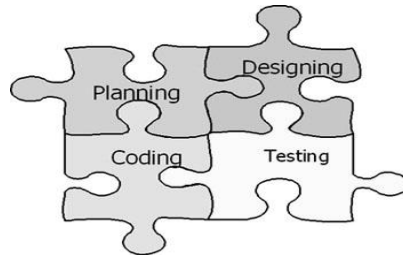


Figura 13 Metodología *Extreme Programming (XP)*

¿Qué se que propone XP?

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio, se convierte en parte sustantiva del proceso.
- El costo del cambio, no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

Características de XP, la metodología se basa en:

- **Pruebas Unitarias:** se basa en las pruebas realizadas a los principales procesos, de manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. De esta manera, es como si nos adelantáramos a obtener los posibles errores.
- **Re-fabricación:** se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto, en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Derechos del Cliente

- Decidir que se implementa
- Saber el estado real y el progreso del proyecto
- Añadir, cambiar o quitar requerimientos en cualquier momento
- Obtener lo máximo de cada semana de trabajo
- Obtener un sistema funcionando cada 3 o 4 meses

Derechos del Desarrollador

- Decidir como se implementan los procesos
- Crear el sistema con la mejor calidad posible
- Pedir al cliente en cualquier momento aclaraciones de los requerimientos
- Estimar el esfuerzo para implementar el sistema
- Cambiar los requerimientos en base a nuevos descubrimientos

Teniendo en cuenta lo antes analizado se puede señalar, que lo fundamental en este tipo de metodología esta dado por: la comunicación entre los usuarios y desarrolladores, la simplicidad al desarrollar y codificar los módulos del sistema y finalmente la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

La experiencia ha demostrado que la clave del éxito de un proyecto de software es la elección correcta de la metodología, que puede conducir al programador a desarrollar un buen sistema de software.

Podemos concluir además, que lo más importante antes de elegir la metodología que usarás para la implementación de tu software, es determinar el alcance que tendrá y luego de ahí ver cual es la que más se acomoda en tu aplicación.

1.4. Gestión de Riesgo

La Gestión de Riesgos en el ámbito del software procura formalizar conocimiento orientado a la minimización o evitación de los mismo en proyectos de desarrollo de software, mediante la generación de principios y buenas prácticas de aplicación realista (ROPPONEN y LYYTINEN, 2000). Logrando producir aquello que el cliente requiere, en el plazo solicitado y ajustado al presupuesto.

1.4.1. ¿Qué es Riesgo?

Según los autores Juan Fuente y Cueva Lovelle, definieron el término Riesgo como:

“Cualquier suceso que pueda afectar negativamente a la marcha del proyecto en el futuro, es asociado de manera inexorable a cualquier actividad que se lleve a cabo y que imponga la decisión entre varias alternativas, por tanto, acompaña todo cambio y está presente en cada decisión” (FUENTE y LOVELLE, 2006).

Una de las personalidades más importantes que tiene la Ingeniería de Software Roger. S. Pressman, cita la siguiente definición acerca del Riesgo, la cual fue planteada por Robert Charrette en su libro Análisis y Gestión de Riesgo.

“En primer lugar, el riesgo afecta a los futuros acontecimientos. El hoy y el ayer están más allá de lo que nos pueda preocupar, pues ya estamos cosechando lo que sembramos previamente con nuestras acciones del pasado. La pregunta es, podemos por tanto, cambiando nuestras acciones actuales, crear una oportunidad para una situación diferente y, con suerte, mejor para nosotros en el futuro. Esto significa, en segundo lugar, que el riesgo implica cambio, que puede venir dado por cambios de opinión, de acciones, de lugares... En tercer lugar, el riesgo implica elección y la incertidumbre que entraña la elección. Por tanto, el riesgo, Como la muerte, es una de las pocas cosas inevitables de la vida”.

Esta definición fue tomada de un curso de Ingeniería de Software III donde hablan acerca de la Ingeniería Informática de la UIB (Universidad Islas Baleares), (RUBIO, 2001), la cual plantea lo siguiente:

El riesgo se halla de forma implícita asociado a toda actividad:

- Todo suceso se ve marcado por las acciones del pasado, ¿Se puede, por tanto, actuar ahora para crear oportunidades en el futuro?
- El riesgo acompaña a todo cambio
- El riesgo implica elección e incertidumbre.

Para las autoras de este trabajo, la definición que brindan Juan Fuentes y Cuevas Lovelle, es la más explícita, porque con breves palabras logran sintetizar lo expuesto por los otros autores y abarca todo lo relacionado con el significado de Riesgo.

1.4.2. Estrategias Reactiva y Pro - activa

El término Riesgo engloba dos estrategias fundamentales que ayudan de una forma u otra a tratar de prevenir que éste ocurra. Estas estrategias son Reactiva y Pro – activa.

Según estudios realizados se plantea que la mayoría de los casos, los equipos de software solamente confían en la Reactiva.

La estrategia reactiva se han denominado humorísticamente "Escuela de gestión del riesgo de Indiana Jones". En las películas, Indiana Jones, cuando se enfrentaba a una dificultad insuperable, siempre decía

"¡No te preocupes, pensaré en algo!". Nunca se preocupaba de los problemas hasta que ocurrían, entonces reaccionaba como un héroe (MURCIA, 2006).

Esta estrategia es aquella donde se reacciona al presentarse los problemas o imprevisto. Este método realiza una evolución de los riesgos cuando este ya se ha producido y se actúa en consecuencia. Puede acarrear consecuencias negativas, las que pueden llegar a poner el proyecto en peligro (FAJARDO, 2003; RUBIO, 2001).

Lo más frecuente es que el equipo de software no haga nada, hasta que algo vaya mal, luego este tiene que apresurarse para corregir el problema, a este método se le conoce como "de bombero" cuando falla, "la gestión de crisis" entra en acción y es cuando en verdad el proyecto entra en peligro (MURCIA, 2006).

La estrategia más inteligente para el control del riesgo es la pro - activa. Con esta se busca una anticipación o predicción de los problemas o imprevistos y se tratan de solucionar con los planes de contingencia. La estrategia pro – activa debe comenzar mucho antes de que se inicien los trabajos técnicos, luego se identifican los riesgos potenciales, se realiza una evolución de la probabilidad e impacto que provocan estos en el proyecto, se priorizan los que hayan hecho mas daño y de ahí se pasa a realizar un plan para la gestión de riesgo (FAJARDO, 2003; MURCIA, 2006).

El primer objetivo es evitar el riesgo, pero como no se puede evitar todos los riesgos el equipo trabaja para desarrollar un plan de contingencia que le permita responder de una manera eficaz y controlada. Debemos añadir que este otro método aplica la técnica de evaluación previa y sistemática de los riesgos y sus posibles consecuencias. Consecuentemente, este tipo de estrategias permite lograr un menor tiempo de reacción ante la aparición de riesgos impredecibles (MURCIA, 2006; RUBIO, 2001).

1.4.3. Tipos de Riesgos

Se han producido extensos debates sobre la definición más ajustada para el riesgo de software y según algunas fuentes que se han consultado, se puede decir que el riesgo siempre implica dos características: (FAJARDO, 2003; MURCIA, 2006; RUBIO, 2001).

- Incertidumbre: Es donde no existe un término definido, puede ocurrir o no. En el caso de los riesgos, no hay un 100% de probabilidad de ocurrencia.
- Pérdida: Si el riesgo llega a convertirse en realidad, pueden ocurrir resultados no deseados o pérdidas.

Cuando se examinan los riesgos es importante medir el nivel de incertidumbre y el grado de pérdida asociado a cada uno. Para hacerlo, se consideran diferentes categorías de riesgos (MURCIA, 2006).

Categorías de Riesgo

- Los riesgos de proyecto: Son aquellos que amenazan el plan del proyecto. Si estos llegan hacerse realidad; pueden provocar un retraso en la planificación temporal y los costos pueden aumentar considerablemente. Además de identificarse otros problemas que están enfocados principalmente en el, personal (asignación y organización), recursos, los clientes, requisitos y el impacto que pueda causar este en el proyecto. Esta amenaza puede implicar, costo por diversas razones, como por ejemplo:
 - Sobre costo por mayor cantidad de horas de trabajo.
 - Multas por incumplimiento.
 - Inversiones para impedir mayores retrasos.
- Los riesgos técnicos: Son los que tienden amenazar la calidad del producto y la planificación temporal. Si este tipo de riesgo se llega a convertir en realidad, la implementación y otras disciplinas como el diseño, prueba y mantenimiento puede llegar ser difícil o imposible.
- Los riesgos de negocio: Ponen en peligro la viabilidad del software a construir. A menudo amenaza el éxito del proyecto o el producto.

Existen cinco tipos de riesgos de Negocio:

1. Riesgo de mercado: Es cuando se construye un producto o sistema excelente, que al final no es de interés de nadie en particular.
 2. Riesgo estratégico: Cuando el producto que se construyó, no encaja en la estrategia comercial de la compañía.
 3. Riesgo de comercialización: Es aquel producto que se construye, y el departamento de ventas no sabe como va a realizar la venta.
 4. Riesgo de Dirección: Perder el apoyo de una gestión experta debido a cambios de enfoque o cambios de personal.
 5. Riesgo de Presupuesto: Perder presupuesto o personal asignado.
- Los riesgos conocidos: Son aquellos que se pueden descubrir después de una cuidadosa evolución del plan, dentro del entorno técnico y comercial en el que se desarrolla el proyecto, además de otras fuentes de información fiables (Ejemplos: fechas de entregas poco realista, falta de especificación de requisitos o de ámbito de software).
 - Los riesgos predecibles: Son aquellos que se extrapolan a otros proyectos, por la experiencia adquirida anteriormente. (Por ejemplo: cambio del personal, mala comunicación, disminución del esfuerzo del personal).

- Los riesgos impredecibles: Pueden ocurrir pero son extremadamente difíciles de identificar.
- A continuación se muestra un ejemplo de los diferentes tipos de riesgos, que anteriormente se explicaron (FUENTE y LOVELLE, 2006).

Tabla 2 Ejemplo de riesgos

Riesgo	Tipo	Clase
Cambio de equipo de trabajo	Proyecto	Riesgo Humano
Cambio de Gestión	Proyecto	Riesgo organizacional
Indisponibilidad del hardware	Proyecto	Riesgo de Herramienta
Cambio de los requerimientos	Proyecto y producto	Riesgo de requerimiento
Retrasos en las especificaciones	Proyecto y producto	Riesgo de estimación
Subestimación del tamaño	Proyecto y producto	Riesgo de estimación
Insuficiencia de las herramientas	Producto	Riesgo de las herramientas
Cambios de la tecnología	Negocio	Riesgo en la tecnología
Competencia de productos	Negocio	Riesgo político Riesgo económico

1.4.4. Métodos generales de resolución de Riesgos

Para contrarrestar los riesgos existen cuatro métodos de resolución de Riesgos, ellos actúan según la actitud que se muestre al frente al Riesgo. Estos cuatro métodos son: eliminación, retención, evitación y transferencia (FUENTE y LOVELLE, 2006).

- **Eliminación del Riesgo:** Se trata de eliminar los factores que inducen el riesgo y con ello eliminar la posibilidad de exposición.
- **Retención del Riesgo:** Muchos de los Riesgos que se identifican en el proyecto, no es posible eliminarlos y por tanto hay que desarrollarle otro tipo de tratamiento. La retención, es la asunción por parte de los responsables del proyecto, de informar la existencia de un riesgo. El mero hecho de acometer un proyecto, es un riesgo que no puede ser evitado y sólo se debe retener.
- **Evitación del Riesgo:** En estos casos hay que identificar los factores que provocan el riesgo y mantenerlos bajo control, para evitar que el mismo provoque sus efectos. Evitar es diferente de eliminar.

En caso del que el riesgo amenazará permanentemente el proyecto, se debe evitar sus efectos.

- Transferencias del Riego: Algunos tipos de riesgos (normalmente poco probables pero muy negativos) pueden ser transferidos a terceros, mediante la contratación de seguros o haciendo contratos en los que el cliente o los proveedores asumen este riesgo y liberan al equipo de proyecto de su gestión.

1.5. ¿Qué es Gestión?

Gestionar es coordinar todos los recursos disponibles para conseguir determinados objetivos, implica amplias y fuertes interacciones fundamentalmente entre el entorno, las estructuras, el proceso y los productos que se deseen obtener (MARECOS, 2001).

1.5.1. ¿Qué es la Gestión de Riesgo?

La Gestión de Riesgo es la práctica compuesta de procesos, métodos y herramienta que posibilita la gestión de los riesgos en un proyecto y que provee de un entorno disciplinado para la toma de decisiones pro-activas en base a determinar constantemente que puede ir mal (riesgos), identificar cuales son los riesgos mas importantes en los cuales enfocarse e implementar estrategias para gestionarlos (SEI, 2004).

Según Connell (1997), “La función de la Gestión de Riesgos del software es identificar, estudiar y eliminar las fuentes de riesgo antes de que empiecen a amenazar la finalización satisfactoria de un proyecto”.

Esta práctica se debe iniciar desde las primeras fases de un proyecto (durante el sondeo de los conceptos) y se desarrolla durante todo el ciclo de vida del producto (hasta la aprobación de producto o proyecto) (CONNELL, 1997).

Siempre que se indague en esta práctica, se debe tener en cuenta la importancia puede aportar la buena ejecución de la misma. Según (ROSENBERG *et al.*, 1999), la Gestión de Riesgo ayuda a evitar desastres, retrabajo y sobre – trabajo, pero es aún más primordial porque estimula a las personas, de situaciones del tipo ganar – ganar. Una buena Gestión posibilita, por tanto, el aprovechamiento óptimo de recursos y provoca, como consecuencia, el aumento de ganancias y la disminución de pérdidas.

Se puede afirmar que los principales objetivos que engloba la Gestión de Riesgo es de identificar, controlar y eliminar las fuentes que provocan los mismos ante de que empiecen a afectar el cumplimiento del proyecto (SEVILLA, 2006).

1.5.2. Historia de la Gestión de Riesgo

La Gestión de Riesgo ha incursionado por varias generaciones y cada una de ellas tiene características diferentes y modelos que lo identifican.

Primera Generación (Casuística):

Esta generación data de principio de los años 80 y está basada en listas “Casuísticas” especiales para proyectos, donde se identifican casos de riesgos y se extrapolan a otros proyectos. No existe una planificación específica.

En esta generación se define los Riesgos Tecnológicos y las Listas de Comprobación de Riesgo.

Segunda Generación (Taxonómica):

Está dada a principios de los años 90. Basada en modelos de procesos y eventos. Dentro de esta generación se pueden incluir:

- Modelo de Boehm.
- Modelo de Hall y su relación con la madurez de SEI – CMM.
- Modelo de Riesgos del SEI.
- Modelo SPR de mejora de capacidad en la Gestión de Riesgo.

Más adelante se explicará estos modelos a más profundidad.

Tercera Generación (Causal):

Esta es la generación actualmente emergente. Arranca con Eurométodo 96, MAGERIT 97, ISPL 98, etc.

Los principales modelos de Gestión de Riesgo propuestos son:

- Modelo MAGERIT de Gestión de Riesgo en Sistemas adaptados a Proyectos (Transición).
- Modelo de eventos de MAGERIT – Proyectos (Transición).
- Modelo *McFarlan* (Transición).
- Modelo *RiskMan* e iniciativa *RiskDriver*.
- Modelo *DriveSPI*.
- Modelo *Eurométodo*.

- Modelo *ISPL*.
- Modelo *Prisa*.

1.5.3. Modelos que esquematizan la Gestión de Riesgo

En la Gestión de Riesgo como anteriormente se había mencionado existen varios modelos que sintetizan cada uno de los pasos que se deben tener presentes a la hora de llevar a cabo la Gestión. Para realizar un proceso más organizado y eficaz.

A continuación se presentan unas series de figuras, donde cada una en particular, esquematiza la Gestión de Riesgos de una forma diferente.

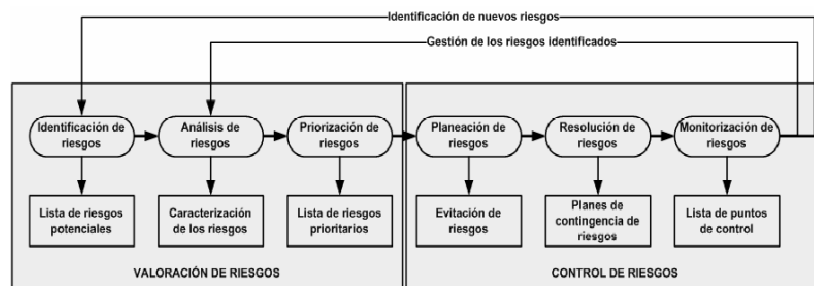


Figura 14 Esquema más detallado del proceso de gestión de riesgo

El modelo anterior muestra los pasos que se pueden tomar como guías para gestionar los riesgos, que se pueden presentar en determinado momento en un proyecto. Este modelo de forma particular, muestra una serie de artefactos que se generan al finalizar cada paso.

Además dividen el proceso de gestión en dos grandes actividades, una la Valoración de los riesgos el cual contiene tres pasos fundamentales y el otro, el Control, que también tiene tres pasos importantes.

Modelo de Gestión de Riesgo según Boehm

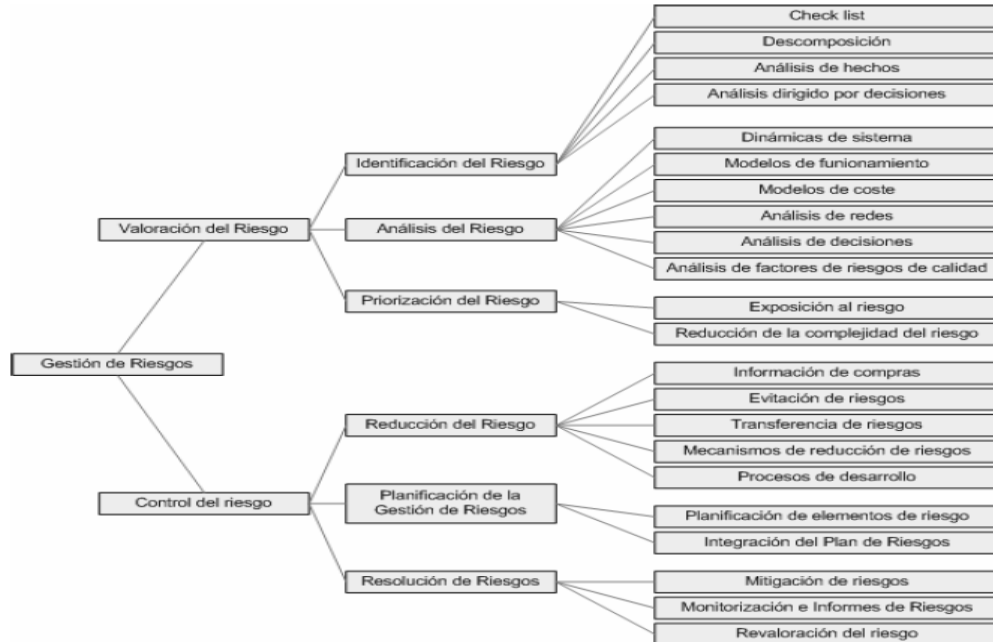


Figura 15 Modelo de Boehm para gestionar los riesgos

Modelo de Gestión de Riesgo en Base a Taxonomías

La Gestión de Riesgos en base a Taxonomías implica, el utilizar una estructura agrupadora de los mismos de acuerdo a sus diferentes clases, como una lista de consultas durante la actividad de Identificación de Riesgos (esta lista estructurada puede obtenerse tanto de la misma organización como de cualquier otra fuente disponible tales como: (MANIASI, 2006).



Figura 16 Modelo de gestión de riesgo en Base a Taxonomía

Modelo de Gestión de Riesgo SQAS – SEI



Figura 17 Modelo de gestión de riesgo SQAS – SEI

El modelo del SQAS – SEI tiene un conjunto de pasos, que ayudan a seguir un orden lógico para la Gestión de Riesgos. Este modelo contiene una tabla, la cual sintetiza una serie de actividades que están dentro de los pasos, las tareas donde se explica la función de la actividad y los artefactos de salida que genera cada paso al terminarse.

Tabla 3 Resumen de los pasos de la gestión de riesgo

	SRM Modelos de Funciones	Pasos de Proceso de las Actividades	Tareas	Salidas	
Comunicación	Identificación	Identificación de Riesgos	El equipo SRE analiza una taxonomía de las áreas para identificar los riesgos potenciales específicos del proyecto	Formulario Inicial de Riesgos	
	Análisis	Análisis de Riesgos	El equipo SRE termina la observación de la forma de la gerencia de riesgo que afectan: coste, horario, calidad de productoy probabilidad, y un consenso del alcance de cada riesgo	Formulario Actualizado de Riesgos	
	Planificación	Priorizar los Riesgos	Un nivel del riesgo se calcula para cada porción del riesgo como los medios de alinear los riesgos del proyecto	Lista de Riesgos Priorizados	
		Identificar los métodos de Aversión de los Riesgos	El equipo SRE realiza un análisis para determinar las acciones de aversión que pueden ser tomadas: Evitación, Control, Asumir y Transferir	Plan de Administración de Riesgos	
		Identificar los métodos de Mitigación de los Riesgos	El equipo de SRE analiza cada riesgo restante y desarrolla acciones de mitigación para reducir la probabilidad y/o la severidad del impacto de cada acontecimiento del riesgo	Plan de Administración de Riesgos	
		Identificar los métodos de Recuperación de los Riesgos	Para cada uno de los riesgos de las etapa N, el equipo de SRE documenta las acciones de contingencia y determinan el disparador para la invocación de una acción de contingencia	Plan de Administración de Riesgos	
		Definir métricas de Riesgos	El equipo de SRE identifica y documenta las métricas necesarias para determinar si se está evitando o se está atenuando un riesgo.	Métricas de Supervisión de Riesgos	
		Implementación de Acciones de Mitigación	La gerencia proactiva procede a poner las acciones identificadas de la aversión en ejecución y de la mitigación de riesgo	Plan de Proyecto Actualizado	
		Seguir	Seguir el Riesgos	Los líderes de proyecto recogen, analizan, y divulgan métricas sobre una base periódica y acontecimiento conducida a la gerencia en cuestión	Programa Activo de Métricas de Riesgos
		Control	Implementación de Acciones de Contingencia	Activar la acción apropiada de contingencia en la identificación del valor métrico señalado por medio de una bandera del riesgo.	Plan de Proyecto Actualizado

Todos los modelos que se presentaron anteriormente tienen un conjunto de pasos o elementos (se le puede decir de las dos formas) que son comunes en cada uno. En algunos casos, los nombres de los elementos no coinciden, además de que algún modelo tenga más pasos que otro. A continuación se muestra el orden en que se pueden aplicar los mismos:

- Identificación
- Análisis
- Priorización
- Planificación
- Resolución y/o Mitigación
- Monitoreo

1.5.4. Explicación detallada de cada uno de los Pasos

Luego de haber mostrado diferentes modelos, que reflejan la continuidad que se debe seguir para la realización de la Gestión de Riesgos, se puede proceder a explicar detalladamente cada una de las funciones, sus características principales y algunos de los métodos o herramientas que sirven de ayuda para la realización de los mismos.

- Identificación de Riesgos

La identificación de Riesgos en proyectos consiste, en la determinación de elementos de riesgos potenciales mediante, la utilización de métodos consistentes y estructurados. Este es, probablemente, el paso más importante entre todos aquellos que componen la Gestión de Riesgos, ya que sin la correcta determinación de los mismo, no es posible desarrollar e implementar anticipadamente problemas que pueden surgir en el proyecto (FUTRELL, 2002).

El Instituto de Ingeniería de *Software* (SEI) supone que la Identificación debe contener todos estos aspectos:

- Métodos sistemáticos y repetibles, para garantizar una gestión consistente.
- Debe ser efectiva, considerar todas las áreas clave de desarrollo y soporte del proyecto.
- Este proceso debe crear un entorno que permita presentar todas las opiniones.
- No se debe hacer un juicio general sobre el éxito o fracaso de un proyecto, basado únicamente en le número o naturaleza de los riesgos no cubiertos.

El tratar de identificar riesgos es un criterio pro - activo, que busca posibles factores de peligro y tomar medidas de aseguramiento o planes de contingencia para contrarrestarlos a ellos y a sus efectos.

Este elemento es un intento sistemático para especificar las amenazas al plan del proyecto (estimaciones, planificación temporal, carga de recursos, entre otros). Identificando los riesgos conocidos y predecibles, el gestor del proyecto da un paso adelante para evitarlos cuando sea posible y controlarlos cuando sea necesario.

Un método para identificar los riesgos es crear una Lista de Comprobación de Elementos de Riesgo, que se define como, un conjunto de cuestiones que son relevantes para cada factor de riesgo. Ver Anexo I.

Este método contiene dos categorías principales (FAJARDO, 2003; RUBIO, 2001; SEVILLA, 2006):

1. Los específicos de producto: Solo se pueden identificar si se tiene una clara visión de la tecnología, el personal y el entorno específico del proyecto en cuestión. Son riesgos asociados a un producto en particular de forma única y específica. Para realizar ésta identificación, se examina el plan del proyecto y la declaración del ámbito del software.
2. Los Riesgos Genéricos: Son una amenaza potencial para todos los proyectos de software.

Tanto los riesgos genéricos como los específicos del producto se deberían identificar sistemáticamente. Tom Gilb tiene toda la razón cuando cita: "Si no atacas activamente a los riesgos. Ellos te atacarán activamente a ti" (MURCIA, 2006).

Los riesgos genéricos se dividen en varias subcategorías, que ayudan a realizar una identificación más rápida de lo riesgos, pueden realizar la función de listas de comprobación de elementos de riesgos. Aclarar que pueden existir otras subcategorías más (FAJARDO, 2003; RUBIO, 2001; SEVILLA, 2006):

- Tamaño del producto: Son los asociados con el tamaño general del software a construir o a modificar, como por ejemplo:
 - ✓ Tamaño estimado del proyecto (LOC/PF).
 - ✓ Confianza en la estimación.
 - ✓ Numero de programas, archivos y transacciones.
 - ✓ Tamaño relativo al resto de proyectos.
 - ✓ Tamaño de la base de datos.
 - ✓ Numero de usuarios.
 - ✓ Numero de cambios de requerimientos previstos antes y después de la entrega.

- ✓ Cantidad de software reutilizado.
- Impacto en el negocio: Aquellos que se asocian con las limitaciones impuestas por la gestión o por el mercado:
 - ✓ Efecto del producto en la cifra de ventas.
 - ✓ Visibilidad desde la dirección de la organización.
 - ✓ Fecha límite de entrega razonable.
 - ✓ Numero de clientes que usarán el producto.
 - ✓ Numero de productos con los que se 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.
- Características del cliente: Son los que están asociado con la sofisticación del cliente y la habilidad del desarrollador, para comunicarse con el cliente en los momentos oportunos.
- Definición del proceso: riesgos asociados con el grado de definición del proceso del software y por su seguimiento la organización de desarrollo.
- Entorno de desarrollo: Aquellos que se vinculan con la disponibilidad y calidad de las herramientas que se van a emplear en la construcción del producto.
- Tecnología a construir: Son los que tienen que ver con la complejidad del sistema a construir y la tecnología que se va a utilizar.
- Tamaño y experiencia del equipo: riesgos asociados con la experiencia técnica y de proyectos de los ingenieros del software que van a realizar el trabajo.

A continuación se muestra un ejemplo de una lista de comprobación de elementos de riesgos, en este caso es sobre el personal (FUENTE y LOVELLE, 2006).

- ¿Se dispone del mejor personal?
- ¿Tiene el personal un conjunto de habilidades adecuado?
- ¿Se dispone del personal suficiente?
- ¿Está comprometido el personal a lo largo de todo el proyecto?
- ¿Hay miembros del equipo que trabajarán sólo a tiempo parcial?
- ¿Se ha creado el personal las expectativas correctas sobre el trabajo que va a realizar?

CAPITULO I. FUNDAMENTACION TEORICA

- ¿Ha recibido el personal la formación adecuada?
- ¿Será suficientemente baja la rotación del personal para permitir la continuidad?

Estas listas tienen ventaja y desventaja (PMI, 2000).

Ventaja: Permiten una identificación de riesgo rápida y relativamente sencilla.

Desventaja: Es prácticamente imposible tener una lista que incluye todos los riesgos en un proyecto software. Se resuelve utilizando la lista como punto de partida, pero permitiendo después incluir varios riesgos específicos del proyecto.

Según Antonio Navarro (2000) en un artículo que publicó sobre la Gestión de Riesgo, planteo una lista que incluye Boehm en su libro Análisis y Gestión de Riesgo, de los 10 riesgos que para él son significativos y a su lado las técnicas que se pueden llevar a cabo para cada uno de ellos (NAVARRO, 2000).

Tabla 4 Lista de los 10 riesgos más frecuentes

Elementos de riesgo	Técnica de gestión de riesgo
1. Deficiencia del personal	Contratar personal con talento, asignación de trabajos, construcción de equipos, acuerdos entre el personal clave, formación cruzada.
2. Planificaciones y presupuestos poco realistas	Estimación multifuente detallada de costes y planificación, diseñar en función del coste, desarrollo incremental, reutilización del software, fregado de requisitos
3. Desarrollo de las funciones y propiedades erróneas	Análisis de organización, análisis de la misión, revisiones del usuario y participación del usuario, prototipado, manuales de usuarios preliminares, formulación de operaciones – conceptos, análisis de rendimientos sin nombre, análisis de calidad – factor.
4. Desarrollo erróneo del interfaz de usuario.	Prototipado, escenarios, análisis de tareas, participación del usuario.
5. Chapado	Fregado de requisitos, prototipado, análisis de costes – beneficios, diseñar en función del coste.
6. Continúa corriente de los cambios en los requisitos.	Umbral de alto cambio, ocultación de información, desarrollo incremental.
7. Deficiencias en componentes proporcionados externamente.	Análisis técnicos, análisis coste – beneficio, prototipado, comprobaciones por referencias
8. Deficiencias en tareas desarrolladas externamente	Benchmarking, inspecciones, comprobaciones por referencia, análisis de la competitividad.
9. Deficiencias en rendimiento en tiempo real.	Comprobaciones por referencias, auditorías ante de los incentivos, contratos con incentivos, diseño o prototipado competitivo, construcción de equipos.
10. Exprimir las capacidades informáticas	Simulación, Benchmarking, modelado, prototipado, instrumentación, ajuste.

Métodos de Identificación de Riesgos.

Existen una gran variedad de métodos, que pueden ser utilizados para identificar los riesgos involucrados en un proyecto, alguno de ellos (ADUANAS, 2005):

1. Discusiones de grupo o entrevistas
2. Experiencia personal del funcionario.
3. Las inspecciones físicas y auditorias.
4. *Brainstorming*.
5. Técnicas Delphi.
6. Encuestas y cuestionarios.
7. Estudio de la experiencia extranjera o nacional.
8. Los juicios, los consensos especulativos, conjeturas, intuiciones.

Brainstorming

Se puede clasificar en varias categorías como son (FUENTE y LOVELLE, 2006):

- Alto Nivel: Normalmente reuniendo a ejecutivos del proyecto, identificando fricciones y riesgos propios de la administración y gestión del proyecto y de las propias políticas organizativas de las empresas implicadas en el proyecto.
- Medio Nivel: reuniones con la Dirección Técnica del Proyecto, identificando riesgos inherentes a las tecnologías, a la organización y comunicaciones con proveedores, etc.
- Bajo Nivel: Reunión con los analistas diseñadores, programadores, etc. Identificando riesgos relacionado con la falta de medios, problemas organizativos, falta de dirección.
- Verticales: En los que se reunirán personas de los distintos niveles implicados para identificar riesgos de comunicación vertical.
- Uso de experiencias anteriores y de bibliografías: Al efecto para identificar algunos riesgos ocultos, pero que se repiten como anti patrones en diferentes proyectos.

Las Técnicas Delphi:

Consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes (ASTIGARRAGA, 2006).

CAPITULO I. FUNDAMENTACION TEORICA

Por lo tanto, la capacidad de predicción de la Delphi se basa en la utilización sistemática de un juicio intuitivo emitido por un grupo de expertos.

Este método presenta tres características fundamentales (ESTÉVEZ y GALLASTEGUI, 2002):

Anonimato: Durante un Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:

- Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
- Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
- El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- Iteración y realimentación controlada: La iteración se consigue al presentar varias veces el mismo cuestionario. Como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.
- Respuesta del grupo en forma estadística: La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

De manera resumida los pasos que se llevarán a cabo para garantizar la realidad de los resultados, el Método Delphi deberá seguir los siguientes pasos (ASTIGARRAGA, 2006):

- Fase 1: Formulación del problema.
- Fase 2: Elección de expertos.
- Fase 3: Elaboración y lanzamiento de los cuestionarios (en paralelo con la fase 2).
- Fase 4: Desarrollo práctico y explotación de los resultados.

CAPITULO I. FUNDAMENTACION TEORICA

- Análisis de Riesgos

Es el proceso donde se examinan los riesgos en detalle para determinar su extensión, sus interrelaciones y su importancia. En este análisis se determina la probabilidad, impacto y exposición o marco de tiempo de cada uno de los riesgos. El equipo que realiza este proceso de tener una actitud pro - activa

Las actividades básicas que están dentro del análisis (FUENTE y LOVELLE, 2006; SEVILLA, 2006):

- Análisis detallado: Mejor comprensión del riesgo, se cuantifican, en lo posible según los siguientes conceptos, esto no son mas que **Atributos** que se encuentran dentro del análisis de Riesgos (Impacto o Consecuencia, Probabilidad, Marco de Tiempo):

Impacto o Consecuencia: Pérdida que ocasiona el riesgo. Consecuencias de los problemas asociados con el riesgo. Estos son los factores que pueden afectar el riesgo:

1. La naturaleza: Problemas potenciales que se pueden producir en caso de ocurrir.
2. Alcance: Combina la severidad (cómo de grave) con su distribución global.
3. Duración: Combina el momento en el que se sentirá su impacto y la duración del mismo.

Probabilidad: Probabilidad de que ocurra el riesgo.

Marco de Tiempo: Período de tiempo en el que es posible mitigar el riesgo.

- Clasificación: Se clasifican los riesgos para entender su naturaleza y elaborar planes de mitigación.

A continuación se muestra una tabla de resumen la cual muestra los atributos, categorías que se mencionaron antes y una pequeña descripción sobre estas categorías (FUENTE y LOVELLE, 2006):

Tabla 5 Resumen de los atributos de los riesgos

Atributo	Valor	Descripción
Consecuencia o Impacto	Catastrófico	Perdida del sistema. Coste mayor del 50%
	Crítico	Recuperación de la capacidad operativa. Coste mayor del 10% y menor del 20%
	Tolerable	Capacidad operativa mermada. Coste mayor del 10% y menor del 20%
Probabilidad	Muy baja	Menor del 10%
	Baja	Del 10% al 22%
	Moderada	Del 25% al 75%
	Alta	Del 75% al 90%
	Muy alta	Mayor del 90%
Marco de Tiempo	Improbable	Menor del 30%
	Corto Plazo	30 días
	Mediano Plazo	De 1 a 4 meses.
	Largo Plazo	Más de 4 meses.

CAPITULO I. FUNDAMENTACION TEORICA

Luego de haber determinado a cada riesgo sus atributos, se debe proceder a decretar el nivel. Para eso existen tres categorías (ADUANAS, 2005):

- Cualitativos.
- Semi – Cualitativos.
- Cuantitativos.

El análisis cualitativo se puede utilizar cuando el nivel de riesgo no justifica el tiempo y los recursos necesarios para hacer el análisis completo.

Un enfoque semi – cuantitativo puede utilizar clasificaciones de palabras como alto, medio o bajo, o descripciones mas detalladas de la probabilidad y la consecuencia.

El método cuantitativo es donde se calcula el nivel de riesgo en situaciones donde la probabilidad de ocurrencia y las consecuencias pueden ser cuantificadas.

En la siguiente tabla se muestra cuales son los posibles métodos de análisis de riesgos (ADUANAS, 2005).

Tabla 6 Métodos de análisis de riesgos

Los métodos cuantitativos incluyen	Los métodos cualitativos incluyen
El análisis de probabilidad.	Brainstorming.
El análisis de consecuencias.	Cuestionario y entrevistas estructuradas
El modelamiento o simulación computacional.	La evaluación que utiliza grupos multi-disciplinarios.
El análisis estadístico / numérico.	El juicio de especialistas y expertos como la técnica de Delphi.
La investigación de mercado.	
Análisis de redes.	
Análisis de costo del ciclo de vida.	
Árboles de decisión.	
Los árboles de falta y los análisis de árbol de eventos.	
Diagramas de influencia.	

El método que se usa con más frecuencia en la toma de decisiones, es el cualitativo. Los administradores utilizan la experiencia, el juicio, la intuición para tomar estas decisiones. El nivel de los riesgos esta determinado por la relación entre la probabilidad y la consecuencia que usualmente se determina en una tabla (ADUANAS, 2005).

Tabla 7 Niveles de los riesgos

Consecuencias

	Extremo	Muy Alto	Medio	Bajo	Mínimo
Casi Cierto	Alto	Alto	Alto	Importante	Importante
Probable	Alto	Alto	Importante	Importante	Significativo
Moderado	Alto	Alto	Importante	Significativo	Bajo
Poco Probable	Alto	Importante	Significativo	Bajo	Bajo
Casi Improbable	Importante	Importante	Significativo	Bajo	Bajo

Probabilidad

Riesgo Alto: Se requiere una investigación detallada y una planificación a niveles superiores.

Riesgo Importante: Se requiere una atención del personal superior.

Riesgo Significativo: Se debe especificar la responsabilidad de gestión.

Riesgo Bajo: Se maneja mediante procedimientos de rutina.

Una técnica razonable que se utiliza en este paso, es desarrollar una tabla de para saber la proyección del riesgo. Los pasos generales serian:

1. Hacer un listado de todos los riesgos posibles sin importar lo remotos que sean.
2. Categorizar cada riesgo, indicando que tipo de riesgo es, si es relacionado con el tamaño de producto, tipo de cliente, entre otros más.
3. Determinar la probabilidad de ocurrencia de cada uno de los eventos planteados, ya sea en forma objetiva o subjetiva, y se puede hacer una valoración promediada de todas las estimaciones individuales hechas para cada evento.
4. Se valora el impacto de cada riesgo y se establece una categoría de impacto. Dichas categorías son:
 - Catastrófico.
 - Crítico.
 - Tolerable.

5. Se realiza una ordenación en función de la probabilidad y el impacto quedando en la parte superior las de mayor probabilidad y mayor impacto.
6. Se establece una línea de corte a partir de la cual se establece que eventos serán de verdadera atención y control.
7. Se establece un plan RSGR que es un *Plan de Reducción, Supervisión y Gestión de Riesgo*, que se busca con este plan
 - Evitar el riesgo.
 - Supervisar el riesgo.
 - Gestión de riesgo y plan de contingencia.

- Priorización de Riesgos

Luego del análisis, este paso trata de decidir si los riesgos son aceptables o no, esto se realiza comparando el nivel que tiene cada uno.

De esta manera se clasifican los riesgos más importantes, y se le da la prioridad que le corresponde, según la repercusión que este pueda tener en el proyecto.

La evaluación debe tomar en cuenta el grado de control sobre cada uno de los riesgos y el impacto de costos, los beneficios y las oportunidades presentadas por los riesgos (ADUANAS, 2005).

- Planificación de Riesgo.

La función de la planificación, es convertir la información de los riesgos en decisiones y acciones para el presente y el futuro.

En esta actividad se construye un Plan de Contingencia, para cada uno de los riesgos identificados en las actividades anteriores y que se han elegido como significativos para gestionarlos.

La colección de todos los planes de contingencia se suele agrupar en el llamado Plan de Respuestas a Riesgos. Este plan incluye (PMI, 2000):

- Riesgos identificados, sus descripciones, aspectos del proyecto afectado (elementos del alcance), sus causas, efectos en los objetivos del proyecto.
- Responsabilidades asignadas en cuanto a cada riesgo.

- Resultados del análisis de riesgos (probabilidad, impacto, exposición del riesgo), incluyendo su priorización.
- Planes de Contingencia: Respuesta prevista para aquellos riesgos que se han considerado prioritarios y que serán planificados.
- Nivel de riesgo residual esperado después de que aplique la respuesta prevista.
- Acciones específicas para implementar la estrategia de respuesta a cambios.
- Presupuesto y tiempo para las respuestas.

Considera cada riesgo y desarrolla una estrategia para manejarlo (FUENTE y LOVELLE, 2006):

- Estrategias de evitación.
 - Se trata de minimizar la probabilidad de que el riesgo se presente
- Estrategias de minimización
 - Se trata de reducir el impacto del riesgo en el producto o en el proyecto
- Planes de contingencias.
 - Si el riesgo se presenta, el plan de contingencia se encargará de tratar este riesgo.
 - Palió los efectos negativos del riesgo que por éste ya se han producido.

Dentro de la planeación se incluye la identificación de las acciones de aversión del riesgo ellos son cuatro (NAVARRO, 2000):

- Evitar el riesgo: Elegir una alternativa de menor riesgo.
- Controlar el riesgo: Se decide mitigar el riesgo.
- Asumir el riesgo: Se acepta que el riesgo ocurra.
- Transferir el riesgo: Reducir el riesgo compartiéndolo.

Las técnicas de planeación incluyen listas de comprobación de resolución de riesgos y análisis coste – beneficio de cada actividad.

El **análisis de costo – beneficio** es el balance entre el costo de implantación de las medidas preventivas y el riesgo remanente. Además toma en cuenta la posibilidad de ocurrencia de daños materiales, tanto a la instalación, así como pérdidas de la producción durante los períodos de parada y en la reparación de los daños.

CAPITULO I. FUNDAMENTACION TEORICA

En la siguiente tabla se muestra un ejemplo de cómo se pueden planear los riesgos y que estrategias se pueden realizar (FUENTE y LOVELLE, 2006).

Tabla 8 Ejemplo de cómo planificar los riesgos

Riesgo	Estrategias
Problemas financieros organizacionales	Preparación de un documento para la dirección de la empresa en la que se muestre la importante contribución del proyecto a los objetivos del negocio.
Retrasos en las entregas	Comunicación al cliente de posibles dificultades y de la posibilidad de que se retarse en proyecto. Investigación de los componentes que provocaron el retraso.
Enfermedad de algún componente	Reorganización del trabajo de manera que no haya solapamientos, ni huecos en los trabajos de manera que todos puedan saber sus responsabilidades y conozcan el trabajo de los demás
Componentes defectuosos	Reemplazar los componentes defectuosos con otros comprados (si existen) y que tengas una efectividad reconocida

- Resolución y/o Mitigación de Riesgo.

La resolución de riesgos implementa todas las estrategias que se tomaron en el paso anterior.

Y es donde se efectúan los planes de contingencias, este asume que la evitación ha fallado y que el riesgo se ha producido. Por ello se define las estrategias y acciones a tomar para lograr que los efectos se minimicen. Nunca se podrá reducir a cero el coste del plan de contingencia, ya que él puede implicar algunos costes en si mismo, por los cual se ha de valorar el beneficio que se espera obtener de éste (FUENTE y LOVELLE, 2006; SÁNCHEZ, 1997).

Las técnicas que se utilizan en la resolución son: relajación de requisitos o poda de requisitos (se le conoce de las dos formas), prototipado y desarrollo incremental.

Poda de requisitos: Es cuando en un determinado momento hay que hacer entrega de una primera versión del producto y el jefe del proyecto toma la decisión de hacer una selección de los requisitos mas importante que se tendrán en cuenta para esa versión, y los demás requisitos pues se integraran para la próxima iteración del producto.

Prototipado: Se brinda una breve panorámica acerca de las acciones que se llevarán a cabo para mitigar los riesgos.

Desarrollo incremental: Que consiste en utilizar incrementos para poder aumentar gradualmente el alcance, empezando por las posibilidades más básicas de las aplicaciones y ampliándolas pasito a pasito, incorporando posibilidades adicionales según unas prioridades establecidas previamente (CONTE, 2007).

- Monitoreo de Riesgo.

El monitoreo es una etapa esencial e integral en el proceso de Gestión de Riesgo. Es necesario el monitorear de: los riesgos, la efectividad del plan, las estrategias y el sistema de administración que ha sido establecido para controlar la implementación de los tratamientos de riesgo (ADUANAS, 2005).

Los riesgos necesitan ser inspeccionados periódicamente para garantizar que las circunstancias cambiantes no alteren las prioridades de los mismos. Son muy pocos los riesgos que permanecen estáticos (ADUANAS, 2005).

Una de las técnicas que se pueden utilizar en este paso son (FUENTE y LOVELLE, 2006):

- El seguimiento de los hitos.
- Discutir en cada reunión de avance de proyecto la calve de cada riesgo.

1.5.5. Resumen de los pasos de la Gestión de Riesgo

En la siguiente tabla, realizada por las autoras de este trabajo, mostrará un resumen de cada paso de los que anteriormente se describieron. Se tendrán en cuenta algunos aspectos para la realización de la misma, Como son: Actividades que se llevan a cabo en cada paso, las herramientas que cada uno de ellos utilizan, los artefactos (lo que se quiere decir con artefacto es lo se genera en cada paso.) y los métodos

CAPITULO I. FUNDAMENTACION TEORICA

Tabla 9 Resumen de los pasos

Pasos	Caracterización	Actividades	Artefactos	Métodos
Identificación de Riesgo	<p>Consiste en la determinación de los riesgos potenciales, mediante la utilización de algún método. Este es el paso más importante dentro de la Gestión de Riesgo ya que sin una correcta determinación de los riesgos no es posible desarrollar e implementar anticipadamente problemas que se pueden dar en el Proyecto.</p>	<ol style="list-style-type: none"> 1. Determinar los riesgos potenciales que están afectando al proyecto en estos momentos. 2. Debe desarrollarse un método sistemático y repetible para identificar los riesgos, para que llegue hacer una Gestión de Riesgos consistente. 3. Se debe tener en cuenta todas las áreas claves de desarrollo y soporte del proyecto. 4. Realizar una lista de contenido con los riesgos que se identificaron. 5. Agregarle a cada riesgo su categoría <ul style="list-style-type: none"> - Los específicos de producto. - Riesgo Genéricos <ul style="list-style-type: none"> • Tamaño del producto. • Impacto en el negocio. • Características del producto. • Definición del proceso. • Entorno de desarrollo. • Tecnología a construir. • Tamaño y experiencia de la plantilla 	<p>Lista de Riesgos</p>	<p>Lista de comprobación de Elementos.</p> <p>Discusiones de grupo o entrevistas Experiencia personal del funcionario.</p> <p>Las inspecciones físicas y auditorias anteriores.</p> <p><i>Brainstorming.</i></p> <ul style="list-style-type: none"> • Alto Nivel. • Medio Nivel. • Bajo Nivel. • Verticales. • Uso de experiencias anteriores y de bibliografía <p>Técnicas Delphi.</p> <p>Encuestas y cuestionarios.</p> <p>Estudio de la experiencia extrajera o nacional.</p> <p>Los juicios, los consensos especulativos, conjeturas, intuiciones.</p> <p>Taxonomía basada en riesgos</p> <p>Lista de Chequeo</p>

CAPITULO I. FUNDAMENTACION TEORICA

		6. Realizar listas de comprobación de elementos de riesgos		
Análisis de Riesgo	Es donde se examinan los riesgos más detalladamente, para determinar sus extensiones, sus interrelaciones y su importancia. En este paso es donde se definen los atributos de los riesgos, su impacto, probabilidad y marco de tiempo.	<ol style="list-style-type: none"> 1. Realizar un listado con todos los riesgos que se tienen. 2. Realizar un análisis detallado de cada uno. 3. Se valora un impacto de cada riesgo y se establece una categoría de Impacto. Dichas categorías son: <ul style="list-style-type: none"> • Catastrófico. • Crítico. • Tolerable. 4. Realizar una ordenación en función de la probabilidad y el impacto quedando en la parte superior las de mayor probabilidad y mayor impacto. Las clasificaciones para la probabilidad son: <ul style="list-style-type: none"> • Muy Baja. • Baja. • Moderada. • Alta. • Muy Alta. 	Caracterización de los Riesgos	<p>Elaboración de una tabla de riesgo.</p> <p>Existen 3 métodos importante en este paso(dentro de ellos existen varios métodos mas)ellos son:</p> <ul style="list-style-type: none"> • Cualitativo. • Semi – Cualitativo. • Cuantitativo.

CAPITULO I. FUNDAMENTACION TEORICA

<p>Priorización de Riesgo</p>	<p>Decidir si los riesgos son aceptables o no y de esta forma se llega a una clasificación de los riesgos más importantes.</p>	<p>1. Realizar la decisión si los riesgos van hacer aceptables o no, esto se hace comparando el nivel de cada riesgo. 2. Se realiza la priorización de los riesgos.</p>	<p>Lista de riesgos prioritario</p>	
<p>Planeación de Riesgo</p>	<p>Ayuda a manejar cada elemento de riesgo incluyendo la coordinación de los planes individuales de elementos de riesgo entre ellos y con respecto al plan general.</p>	<p>1. Se realizan las listas de comprobación de resolución de riesgo. 2. Se tiene en cuenta el coste – beneficio. 3. Se desarrolla una estrategia para manejar cada riesgo: <ul style="list-style-type: none"> • Evitación • Minimización • Planes de contingencia. 4. Se identifican las acciones de aversión del riesgo ellos son: <ul style="list-style-type: none"> • Evitar el riesgo • Controlar el riesgo. • Asumir el riesgo • Transferir el riesgo </p>	<p>Evitación de Riesgo</p>	<p>Coste – beneficio.</p>
<p>Resolución y/o Mitigación de Riesgo</p>	<p>Implementa la planeación es decir todo lo que se plantea en este paso, también implementa los planes de</p>	<p>1. Implementar las estrategias que se tomaron en la planeación. 2. Se realiza el plan de contingencia. 3. Poner en práctica</p>	<p>Planes de Contingencia</p>	<p>Relajación de requisito o la Poda de requisito. Prototipado. Desarrollo incremental</p>

CAPITULO I. FUNDAMENTACION TEORICA

	contingencia	<p>las técnicas que tiene este paso:</p> <ul style="list-style-type: none"> • Poda de requisitos. • Prototipado. • Desarrollo incremental. 		<i>Benchmarking</i>
Monitoreo de Riesgo	<p>Una etapa esencial e integral en el proceso de Gestión de Riesgo. Es necesario monitorear: los riesgos, la efectividad del plan, las estrategias y el sistema de administración que ha sido establecido para controlar la implementación de los tratamientos de riesgo.</p>	<p>1. Realizar el monitoreo.</p> <ul style="list-style-type: none"> • Los riesgos. • La efectividad del plan. • Las estrategias • Sistemas de administración que se estableció para controlar la implantación de los tratamientos de riesgo. <p>2. Realizar un control periódico para garantizar que las circunstancias cambiantes no alteren los riesgos.</p>	Lista de puntos de Control	<p>Monitoreo de los Riesgos principales</p> <p>Revisión e Inspección.</p> <p>Reuniones de avance de Proyecto</p> <p>Auditorias</p>

1.5.6. Herramientas usadas para la Gestión de Riesgo

Existen gran cantidad de herramientas de software de Gestión de Riesgo disponible en el mercado y que siguen algunas metodologías.

Estas herramientas se enfocan sólo en una categoría de riesgos (*TRIMS – Technical Risk Identification and Mitigation System*), o están orientadas a compañías maduras que poseen una amplia base de datos organizacional que les permita generar información de categorías propias de riesgos (*Risk Trak* y *WelcomRisk*), o bien emplean un mecanismo que no se orienta al uso de taxonomías (*ARM – Active Risk Manager*) (FUENTE y LOVELLE, 2006).

Tabla 10 Resumen de herramientas

Producto	Proveedor	Descripción	Plataforma
<i>Active Risk Manager (ARM)</i>	<i>Strategic Thought</i>	Herramienta integrada de Administración de Riesgos que brinda una solución para la Identificación de Riesgos mediante la utilización de la información contenida en la WBS de proyecto.	<i>Web Based</i>
<i>Technical Risk And Mitigation System (TRIMS)</i>	<i>Best Manufacturing Practices</i>	Herramienta integrada de Administración de Riesgos que emplea ingeniería de conocimiento y que se enfoca en la Identificación y medición de riesgos técnicos de proyecto	<i>Win 32</i>
<i>RiskTrak</i>	<i>Risk Services & Technology</i>	Herramienta integrada a la Administración de Riesgos que brinda una solución para la Identificación de Riesgos mediante el empleo de bases de datos	<i>Win 32</i>
<i>WelcomRisk</i>	<i>Welcom</i>	Herramienta integrada a la Administración de Riesgos que brinda una solución para la Identificación Sistemática de Riesgo mediante la utilización de bibliotecas configurables de categorías de riesgo.	<i>Win32</i>
<i>Chichón – Análisis de Riesgo</i>	<i>Free</i>	Chichón es una herramienta para analizar cuantitativamente el riesgo de un sistema (de información). La herramienta sigue el Modelo de Magerit 1.0	<i>Java</i>

1.6. Conclusiones

La Gestión de Riesgo en la UCI es importante y necesaria, debido a que se pueden detectar los riesgos que afectan al proyecto en cualquiera de las fases en que esté el mismo, favorece a que el software posea la Calidad que se espera. La Gestión de Riesgo se debe realizar mediante todo el Proceso de Desarrollo del Software, ya que este la incluye dentro de sus actividades protectoras.

Para llevar a cabo una buen manejo de los riesgo se deben tener presente una serie de pasos los cuales se describen detalladamente en uno de los epígrafes de este capítulo

CAPÍTULO 2: PROCESO DE GESTIÓN DE RIESGO EN LOS PROYECTOS PRODUCTIVOS

2.1. Caracterizar la Gestión de Riesgo y el Proceso de Desarrollo de Software en los Proyectos Productivos

Durante el curso 2006-2007, en la UCI, se llevo a cabo la aplicación de encuestas y entrevistas, a una población de 19 personas. Con éste método se logró caracterizar el Proceso de Desarrollo de Software y la Gestión de Riesgos involucrados en los proyectos productivos que se ejecutan en esta institución. Dichas encuestas le fueron aplicadas a directivos de las 10 facultades y algunos líderes de proyecto, que tienen suficiente experiencia en la dirección de los mismos. Ver Anexo III.

La Información obtenida permitió arribar a diversas conclusiones, las que luego desataron una serie de investigaciones.

2.1.1. Caracterizar el Proceso de Desarrollo de Software

La primera conclusión que se obtuvo por respuestas de los encuestados y entrevistados, fue que la metodología mas aplicada en el Proceso de Desarrollo de Software en los proyectos productivos de la Universidad de RUP (*Rational Unified Process*).

¿Por qué RUP y no otra metodología?

Según las investigaciones el 95% de los encuestados y entrevistados usan ésta metodología, porque es fácil de configurar, es estándar y comprende tres principios fundamentales:

- Guiado por Casos de Uso
- Centrado en la Arquitectura
- Iterativo e Incremental

Otro argumento sostenido en la investigación además, es que está dividida en fases, iteraciones y disciplinas y define un conjunto de roles que permiten delegarles responsabilidades todos los miembros del equipo de desarrollo. Se caracteriza también porque es usado en proyectos de gran envergadura, que requieren recursos humanos capaces de administrar un proceso complejo en varias etapas.

2.1.2. Caracterizar la Gestión de Riesgos

La segunda conclusión que arrojó la investigación que se llevó a cabo fue la siguiente: no se realiza una eficiente Gestión de los Riesgos en los proyectos productivos. Este hecho puede traer consigo atraso en el cronograma del proyecto, la calidad del producto no es la requerida, que se entregue en el tiempo estimado y con el costo que asignó. Otra deficiencia encontrada, es que sólo se tiene en cuenta algunos de los pasos que están establecidos para Gestionar los riesgos.

2.1.3. Caracterización de las encuestas

Preguntas 4 y 9.

Siendo RUP, la metodología más, el 89% de los encuestados respondieron que el modelo de mayor aplicación es el Incremental y un 74 % manifestaron que era el Iterativo. Dándose el caso, que muchos de los encuestados marcaban el Iterativo e Incremental y viceversa. De allí, que el 53% de los entrevistados planifiquen las iteraciones. Ver Anexo II.

Preguntas 1 y 6.

En el tema de Gestión de Riesgo, solo el 53% de los encuestados respondió que realizaba dicha gestión. Ese porcentaje no indica que se gestionen de una forma eficiente los riesgos, por la razón de que no poseen una guía, un modelo y orden lógico para ejecutar los pasos. De esta forma se concluye que un 42% de los encuestados realicen reuniones de avance de proyecto para el seguimiento de los riesgos. Ver Anexo II.

2.2. La Administración de Riesgos según el *Rational Unified Process*¹

Rational Unified Process tiene dentro de sus disciplinas la Administración de Proyectos y ésta a su vez incluye la forma en que se deben Administrar los Riesgos en todo el Proceso de Desarrollo de Software.

Esta metodología plantea, que un riesgo en la vida diaria, puede ser una exposición de pérdida o daño a un factor, cosa, elemento a un curso que implica peligro. Pero en el ámbito del desarrollo de software, se puede definir el riesgo como una variable que, dentro de su distribución normal puede tomar valor que ponga en peligro o elimine el éxito del proyecto (RUP, 2003).

Según RUP los riesgos se pueden clasificar en directo e indirectos:

¹ Todo el texto referido en este epígrafe es una traducción al español del original en inglés, por las autoras de este trabajo.

CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

- Riesgo Directo: Es el riesgo sobre el cual el proyecto tiene control.
- Riesgo Indirecto: Es aquel sobre el cual el proyecto tiene poco o nada de control.

En algunas ocasiones un riesgo indirecto puede llegar a ser directo, disfrazado (RUP, 2003).

Esta misma fuente define tipos de riesgos que se agrupan en varias categorías. Estas pueden ser:

- Riesgos de Recursos que su vez se clasifican en de: Organización, Financiero, Tiempo y Personal.
- Riesgos de Negocio.
- Riesgos Técnicos que incluyen a los Riesgos Tecnológicos, de Dependencia y Alcance del Riesgo.

A continuación se muestra una tabla con las interrogantes que permiten identificar los diferentes tipos de riesgos.

Tabla 11 Tipos de Riesgos según RUP

Categoría	Tipo	Interrogante
Riesgos de Recursos	Organización	<ul style="list-style-type: none"> ○ ¿Hay suficiente compromiso para este proyecto (incluyendo: la Administración, los proveedores y otros partidos externos implicados)? ○ ¿Es este el proyecto más grande que ha procurado siempre la organización? ○ ¿Se tiene un proceso bien definido para la Ingeniería de Software? ○ ¿Se realiza una buena Gestión y captura de los requisitos?
	Financiero	<ul style="list-style-type: none"> ○ ¿Está definido de financiamiento completo para el proyecto? ○ ¿Se ha asignado el financiamiento, para el aprendizaje y la tutoría? ○ ¿Hay limitaciones con el presupuesto, por lo cual el sistema deba ser entregado con costo fijo o estar de acuerdo con la cancelación?
	Tiempo	<ul style="list-style-type: none"> ○ ¿Es real el tiempo? ○ ¿Es crítica la fecha de liberación?

CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

	Persona	<ul style="list-style-type: none"> ○ ¿El personal disponible es suficiente? ○ ¿Tienen las habilidades y experiencia apropiada? ○ ¿Es algún momento han trabajado junto? ○ ¿Creen que el proyecto pueda tener éxito? ○ ¿Los representantes por parte de los usuarios están disponibles para las revisiones? ○ ¿Son expertos en el dominio disponible?
Riesgos de Negocio		<ul style="list-style-type: none"> ○ ¿Es mayor el valor del sistema proyectado, que el costo que se estimó? ○ ¿Qué si los contratos no se pueden hacer con los surtidores principales? ○ ¿Qué pasa si un competidor puede llegar alcanzar rápidamente el mercado?
Riesgos Técnicos	Alcance del Riesgo	<ul style="list-style-type: none"> ○ ¿Existen acuerdos para medir el éxito? ○ ¿Las escalas del tiempo de desarrollo del proyecto son cortas o inflexibles?
	Riesgos Tecnológicos	<ul style="list-style-type: none"> ○ ¿Se ha probado la Tecnología? ○ ¿Son razonables los objetivos de la reutilización? ○ ¿Son razonables los volúmenes de transacción? ○ ¿Son creíbles las estimaciones de las tarifas de la transacción? ○ ¿Son demasiado optimista? ○ ¿Existen dependencias externas se interfaces a otros sistemas, incluyendo alguno fuera de la empresa? ¿Las interfaces existen o deben ser creadas? ○ ¿Existen requisitos disponibles y seguros, los cuales son extremadamente inflexibles (por ejemplo: El sistema nunca debe fallar)?

2.2.1. Conceptos fundamentales que brinda RUP sobre La Administración de Riesgos

La Administración de Proyectos tiene entre sus actividades Identificar y Evaluar Riesgos. Estos son realizados por el líder de proyecto y tiene como propósito:

- Identificar, analizar y priorizar los riesgos del proyecto y determinar estrategias apropiadas para la Gestión de Riesgos.
- Actualizar la lista de riesgos para reflexionar sobre el estado actual del proyecto.

Dentro de la Administración de Proyectos esta contenida la Administración de Riesgos. Para llevar a cabo esta última se debe tener en cuenta los siguientes pasos (RUP, 2003):

1) Definir el procedimiento y las herramientas de Administración de Riesgos

En este paso se deben establecer los procedimientos que se seguirán para Gestionar los riesgos:

- Identificar los riesgos
- Analizar los riesgos
- Priorizar los riesgos

También se deben identificar las herramientas o técnicas especializadas que se van a utilizar para capturar y almacenar la información del riesgo.

2) Crear una lista inicial de Riesgo

Antes de decidir las estrategias que se ven a utilizar, se debe tener conocimiento de los tipos de riesgos que están frecuentando el proyecto en esos momentos. Luego de haber realizado este proceso, se pasa a crear una lista inicial de riesgos, los que van a servir como guía para posteriores pasos.

3) Asignar un equipo de Administración de Riesgos

Se realizará una selección de algunos miembros del proyecto, que serán los responsables de llevar a cabo la Gestión de los Riesgos. En esta selección se debe tener en cuenta el representante por parte de los clientes, el arquitecto de software y líder de los equipos de prueba, desarrollo, documentación y despliegue. Luego se debe proceder a asignar un Jefe para este equipo, que será el responsable de reunir a todo el personal para realizar la identificación y reporte de los riesgos, además de programar las reuniones para monitorear los mismos.

CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

4) Decidir estrategias para la Gestión de los 10 riesgos más frecuentes

Para cada uno de los riesgos que se encuentran en la lista, el equipo de Gestión debe decidir los métodos que se van a utilizar, para mantener un chequeo de los mismos y en caso que ocurra el riesgo, se debe proceder a poner en práctica un Plan de Contingencia. Los métodos que se incluyen son: “la evitación”, transferencia, aceptación y la mitigación.

5) Definir los indicadores para los 10 riesgos

En caso de que cada riesgo que está en la lista ocurra, se debe definir una condición medible, que indique que el mismo se ha convertido en realidad. A estas condiciones son las que se le conocen como indicadores.

El líder de proyecto supervisará los mismos a través del proyecto e implementará el Plan de Contingencia.

6) Determinar un Cronograma

La Administración de Riesgos es más efectiva si se trata como un proceso continuo. Esta debe realizar un cronograma que servirá para la planificación del equipo de Gestión, logrando de esta manera desarrollar los reportes y reuniones de revisiones de los riesgos en tiempo. Además debe quedar plasmado, las veces que se realizaron algunas reuniones que no estaban planificadas.

2.2.2. Pasos que define para la Gestión de Riesgos

Esta disciplina asume como pasos principales a seguir para Gestionar los Riesgos (RUP, 2003).

- Identificar los riesgos potenciales
- Analizar y Priorizar los riesgos
- Identificar estrategias para mitigar riesgos
- Identificar estrategias de Contingencia
- Revisar los riesgos durante la iteración
- Revisar los riesgos al final de la iteración

Identificar los riesgos potenciales:

Este paso tiene como principal propósito, la realización de una inspección para ver que puede ir mal en el proyecto.

RUP brinda una guía para identificar los riesgos potenciales, que comienza por:

CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

1. La realización de una lista de los posibles riesgos en la fase de inicio, que se elabora por el líder del equipo de Gestión.
2. La utilización de varias técnicas que ayudan a la identificación.
 - Tormenta de Ideas (*Brainstorming*)
 - Lista genérica de riesgos, para ello se utiliza: Asignación y Control de riesgos de Software por Caper Jones o la Identificación de Riesgos Taxonómicas, establecidas por el SEI.
3. Continuar la actualización de la lista, fases posteriores.

Analizar y Priorizar los riesgos:

Posee el propósito de combinar los riesgos similares(es decir, reducir el tamaño de la lista) y alinearlos según el impacto de ellos en el proyecto.

Ahora bien, en caso de no haber identificado más ningún riesgo, se debe examinar la lista que fue elaborada en el paso anterior y combinarlos para eliminar los que se encuentran duplicados.

Los riesgos presentan una serie de atributos, que manifiestan la importancia que puede llegar a tomar cada uno en el proyecto.

Estos atributos son:

- Impacto de los riesgos: Son los que influyen en las desviaciones del planeamiento, esfuerzo o costo el cual se destina para el plan, en caso de que el riesgo ocurra. (este atributo se expresa con un porcentaje).
- Probabilidad de ocurrencia: Es la probabilidad de que ocurra un riesgo (se expresa también con un porcentaje).
- Exposición del riesgo: Se calcula mediante la multiplicación del impacto por la probabilidad de ocurrencia (este puede ser expresado en semanas o días).

Identificar de estrategias para mitigar los riesgos:

La intención que propone éste paso, es desarrollar planes de mitigación, para reducir el impacto de los riesgos.

La mitigación se le realiza a los riesgos directos, es decir, aquellos que el proyecto tiene control. Se identifican las acciones a tomar para reducir la probabilidad y el impacto.

Las acciones de mitigación en un proceso iterativo, se debe realizar desde iteraciones tempranas. En caso de que se confirme el riesgo, se responde con un plan de Contingencia.

CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

Identificar de las estrategias de Contingencia:

Esta estrategia se pone en práctica, cuando se materializa un riesgo o cuando ha fallado la acción de evitar, transferir y la mitigación no fue exitosa.

El plan de Contingencia puede ser considerado por:

Tabla 12 Plan de contingencia

Riego	Indicador	Acción
¿Cuál es el riesgo?	<ul style="list-style-type: none">○ ¿Cómo sabrás que el riesgo se ha convertido en una realidad?○ ¿Cómo se reconoce el acontecimiento de pérdida?	¿Qué se debe hacer para solucionar el riesgo?

Los indicadores pueden identificarse por:

- Ser monitoreados usando métricas de proyecto.
- Ser monitoreados basados en los requerimientos del proyecto y los resultados de las pruebas.
- Ser asociados a eventos específicos.

Revisar los riesgos durante la iteración:

Este paso asegura que la lista de riesgos se mantenga actualizada en todo el proyecto.

La valoración de los riesgos, debe ser un proceso continuo, que ocurra en todas las iteraciones del producto; para ello se realiza:

- Una revisión de la lista de riesgos semanalmente y observar que cambios ha sufrido la misma.
- Hacer la lista de los 10 riesgos más frecuentes en el proyecto e insistir en una acción, la cual le de solución.

Revisar los riesgos al final de la iteración:

En este se reenfocan en los objetivos de la iteración con respecto a la lista de riesgos. Específicamente:

- Eliminar los riesgos que fueron mitigados completamente.
- Introducir los riesgos que fueron encontrados.
- Reevaluar la magnitud y reordenar la lista de riesgos.

2.3. Proceso para la Gestión de Riesgo

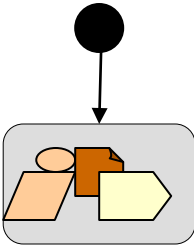
Partiendo de estudios que se realizaron en el Capítulo 1, sobre los modelos que se aplican para la Gestión de Riesgo y basándose en lo que plantea el *Racional Unified Process* en su ayuda extendida, se procede a la realización de una propuesta para la Gestión de Riesgo en la UCI.

El proceso se basa básicamente en la identificación, análisis, priorización, planificación, resolución y/o mitigación y monitoreo de los riesgos en los proyectos productivos.

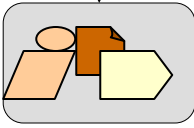
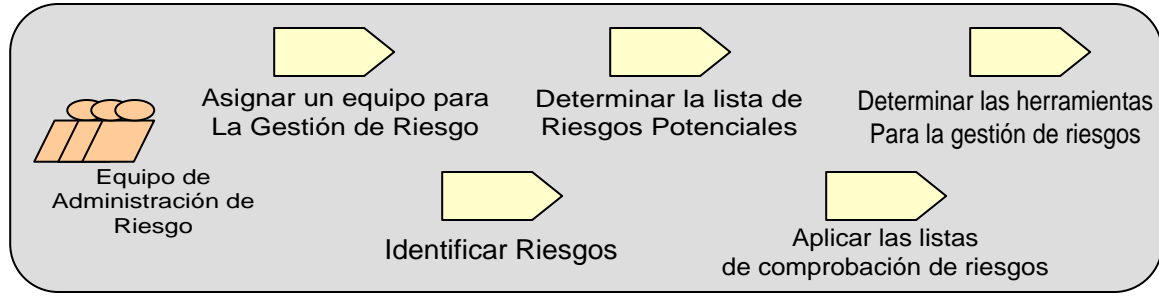
Las autoras de este trabajo definen:

- Para proyectos pequeños no se hace necesario un equipo de administración de riesgo que desarrolle los flujos de trabajo, esto solo lo puede realizar el líder de proyecto.
- Para proyectos de gran envergadura, entra actuar el equipo de administración.

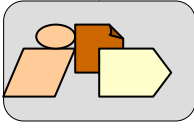
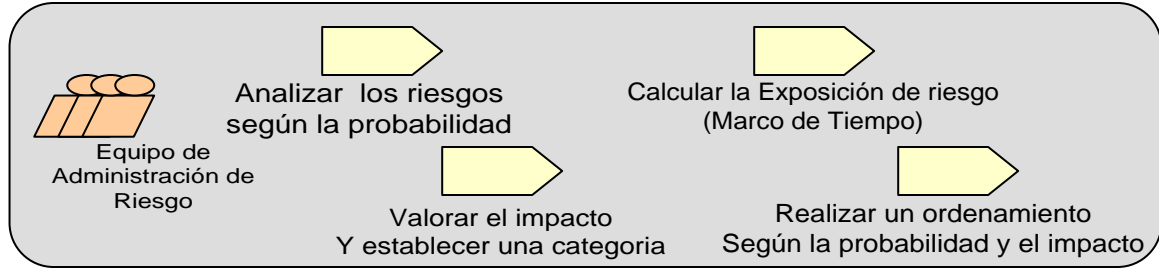
Inicio del Proceso



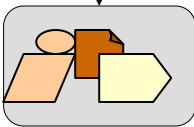
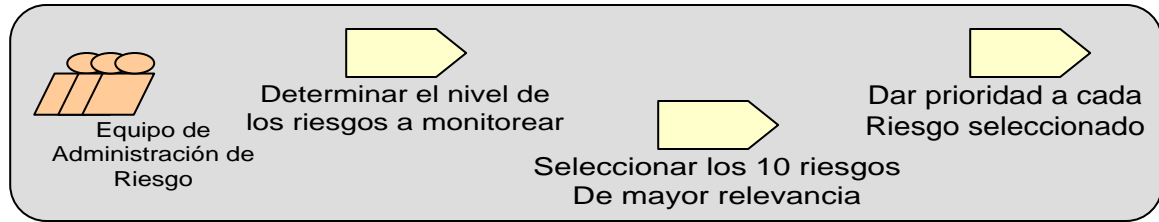
Identificación



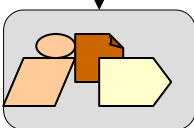
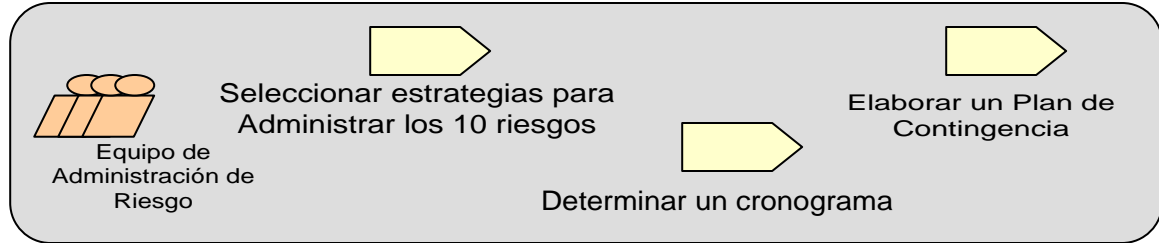
Análisis



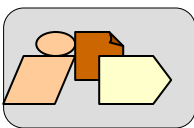
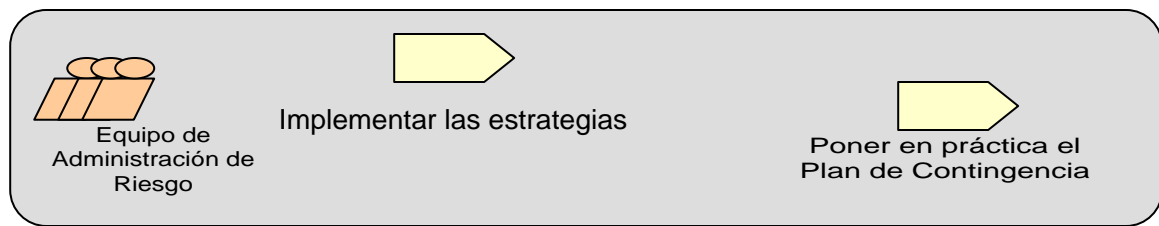
Priorización



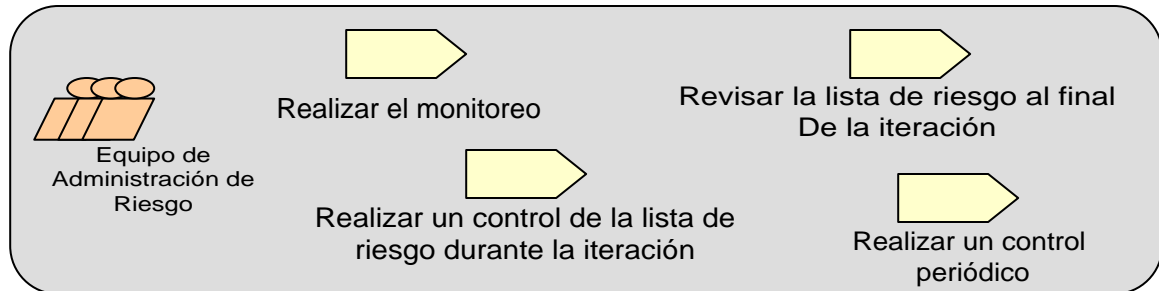
Planificación



Resolución y/o Mitigación



Monitoreo

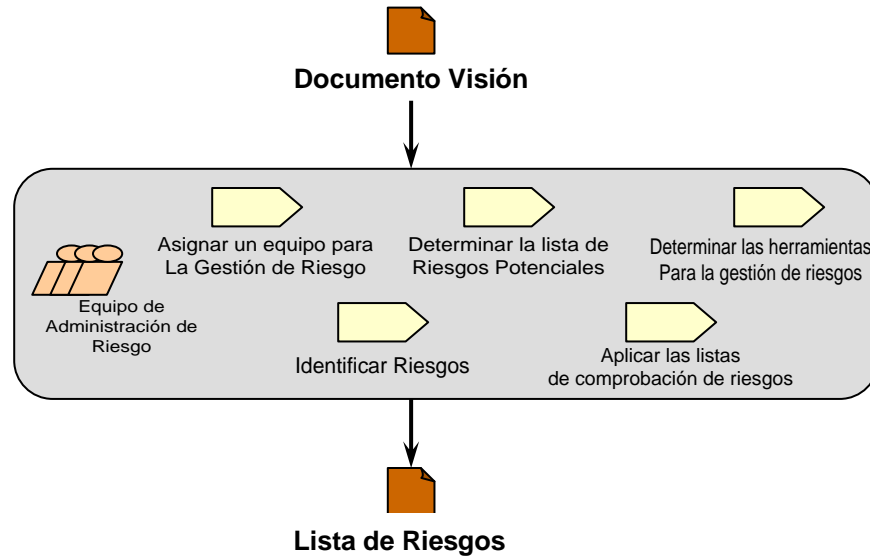


Fin del Proceso

Identificación de nuevos riesgos

2.3.1. Flujo de Trabajo: Identificación

La identificación es uno de los flujos más importantes dentro de la Gestión de Riesgo. Consiste en la determinación de los riesgos potenciales, mediante la utilización de métodos o técnicas. Posibilitando la implementación y desarrollo anticipado de los problemas que pueden surgir en el proyecto.



2.3.1.1. Asignar un equipo para la Gestión de Riesgos

En esta actividad se realiza la creación del equipo para la Gestión de Riesgo. Para conformar este equipo, se debe tener en cuenta el representante por parte de los clientes, el arquitecto de software y el líder de los equipos de prueba, desarrollo, documentación y despliegue, los que se encargarán de transmitir y analizar con todo su equipo los posibles riesgos en su proyecto. Inmediatamente se procede a la selección del líder de este equipo, que será el responsable de reunir al personal para realizar la identificación, el reporte y monitoreo de los riesgos.

2.3.1.2. Identificar los riesgos

El tratar de identificar, es un criterio pro _ activo. Se realiza buscando posibles factores de riesgo que puedan afectar al proyecto, teniendo en cuenta todas las áreas de desarrollo.

2.3.1.3. Determinar la lista de riesgos potenciales

Se basada en la realización de una lista, la cual contiene los riesgos más importantes del proyecto.

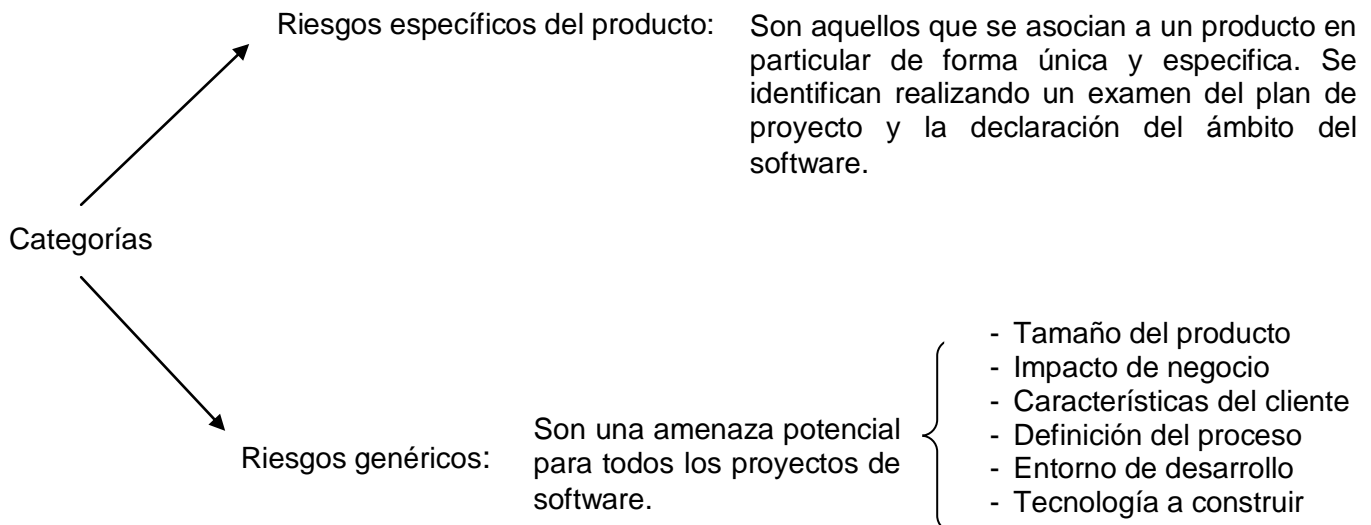
Esta lista se debe actualizar en cada fase, para detectar si se han incorporado nuevos riesgos.

Las técnicas a aplicar para la realización de los dos primeros pasos son:

- Discusiones de grupo o entrevistas
- Experiencia en el personal
- Tormenta de idea (*Brainstorming*)
- Técnicas Delphi

2.3.1.4. Aplicar las listas de comprobación de riesgos

Actividad que define un conjunto de cuestiones que son relevantes para cada factor de riesgo, teniendo en cuenta la categoría de los mismos.



El método que se puede utilizar en este paso son:

- Las Listas de Comprobación de Elementos. Ver Anexo I.

2.3.1.5. Determinar las herramientas para la gestión de riesgos

En esta actividad se exponen un conjunto de herramientas que pueden ser utilizadas para la identificación de riesgos:

CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

- *Active Risk Manager (ARM)*
- *Technical Risk and Mitigation*
- *System (TRIMS)*
- *Risk Trak*
- *Welcome Risk*

(Para más información remitirse al Capítulo 1. Epígrafe “Herramientas usadas para las Gestión de Riesgo”).

Artefacto de entrada:

1. Documento Visión.

Artefacto de salida:

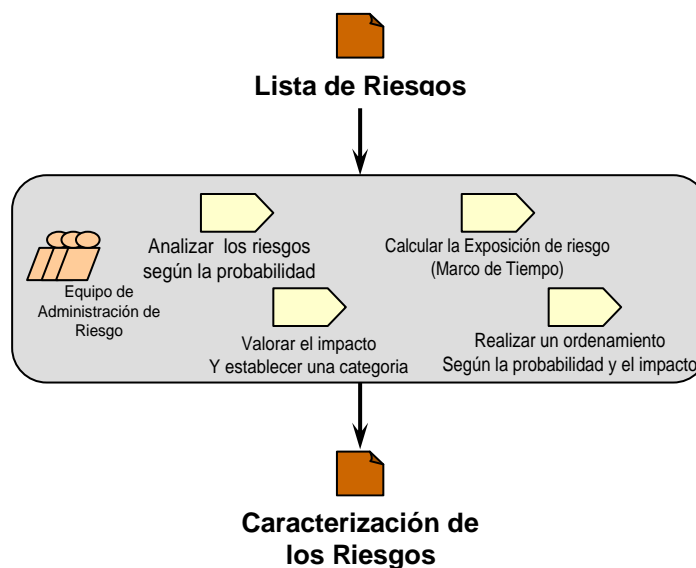
1. Lista de riesgos.

2.3.2. Flujo de Trabajo: Análisis

En este flujo de trabajo se examinan los riesgos con un nivel más detallado, para determinar sus extensiones, interrelaciones e importancia.

Además se definen sus atributos:

- Impacto o Consecuencia
- Probabilidad
- Exposición al riesgo (Marco de Tiempo)



CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

2.3.2.1. Analizar los riesgos según la probabilidad

En la presente actividad se realiza un análisis detallado y una clasificación de los riesgos según la probabilidad de ocurrencia.

La técnica cuantitativa incluye el análisis de la probabilidad, a esta probabilidad se le asigna un parámetro mayor que cero y menor e igual que el cien por ciento. Siendo cero la certeza absoluta de no ocurrencia del suceso y uno cuando se presenta la certeza absoluta de ocurrencia.

Tabla 13 Probabilidad

Atributo	Valor	Descripción
Probabilidad	Muy baja	Menor del 10%
	Baja	Del 10% al 22%
	Moderada	Del 25% al 75%
	Alta	Del 75% al 90%
	Muy alta	Mayor del 90%

2.3.2.2. Valorar el Impacto y establecer una categoría

Esta actividad se basa fundamentalmente en valorar el impacto que puede ocasionar un riesgo en el proyecto.

La técnica a utilizar es el Análisis de Consecuencias, incluida en el método al cual se hizo alusión en el paso anterior.

Tabla 14 Consecuencias o impacto

Atributo	Valor	Descripción
Consecuencia o Impacto	Catastrófico	Pérdida del sistema. Costo mayor del 50%
	Crítico	Recuperación de la capacidad operativa. Costo mayor del 10% y menor del 20%
	Tolerable	Capacidad operativa mermada. Costo mayor del 10% y menor del 20%

2.3.2.3. Calcular la Exposición de riesgo (marco de tiempo)

Se determina el tiempo que va a estar el riesgo en el proyecto. Calculándose mediante la multiplicación de la probabilidad por el impacto.

Los valores de la Exposición al riesgo en un proyecto se muestran en la siguiente:

Tabla 15 Exposición al riesgo o marco de tiempo

Atributo	Valor	Descripción
Exposición al riesgo O Marco de tiempo	Improbable	Menor que 30 días
	Corto Plazo	30 días
	Mediano Plazo	De 1 a 4 meses.
	Largo Plazo	Más de 4 meses.

2.3.2.4. Realizar un ordenamiento según la Probabilidad y el Impacto

Se realiza un ordenamiento de los riesgos, teniendo en cuenta la probabilidad de aparición en el tiempo y el impacto en el proyecto, logrando de esta manera una visión clara de la magnitud de cada riesgo.

Artefacto de entrada:

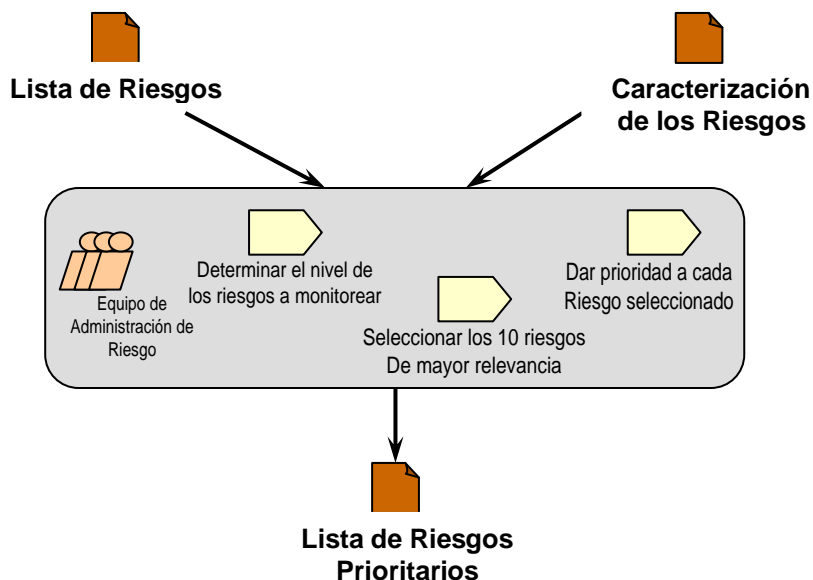
1. Listas de riesgos

Artefacto de salida:

1. Caracterización de los riesgos

2.3.3 Flujo de Trabajo: Priorización

En este flujo se toma la decisión de cuales son los 10 riesgos que tiene mayor relevancia en el proyecto, y se realiza una clasificación para darle una prioridad a cada uno de ellos.



CAPITULO II. PROCESO PARA LA GESTION DE RIESGO

2.3.3.1 Determinar el nivel de los riesgos a monitorear

Calcular el nivel de cada riesgo analizado, que se determina por la relación entre la Probabilidad y la Consecuencia.

Teniendo en cuenta que los atributos de las tablas de equivalencia (16 y 17) que se muestran a continuación, no coinciden con los que se hacen alusión en el flujo anterior, las autoras de este trabajo decidieron realizar un equivalente entre los atributos de ambas tablas.

Tabla 16 Equivalencia para la probabilidad

Casi Improbable	↔	Muy Baja
Poco Probable	↔	Baja
Moderado	↔	Moderado
Probable	↔	Alta
Casi Cierto	↔	Muy Alta

Tabla 17 Equivalencia para la consecuencia

Extrema	↔	Catastrófico
Muy Alto	↔	
Medio	↔	Critico
Bajo	↔	Tolerable
Mínimo	↔	

Para una mayor organización se aconseja consultar la tabla que viene a continuación:

Tabla 18 Consecuencias y Probabilidades

		Consecuencias				
		Extremo	Muy Alto	Medio	Bajo	Mínimo
Probabilidad 	Casi Cierto	Alto	Alto	Alto	Importante	Importante
	Probable	Alto	Alto	Importante	Importante	Significativo
	Moderado	Alto	Alto	Importante	Significativo	Bajo
	Poco Probable	Alto	Importante	Significativo	Bajo	Bajo
	Casi Improbable	Importante	Importante	Significativo	Bajo	Bajo

- Riesgo Alto: Se requiere una investigación detallada y una planificación a niveles superiores.
- Riesgo Importante: Se requiere una atención del personal superior.
- Riesgo Significativo: Se debe especificar la responsabilidad de gestión.
- Riesgo Bajo: Se maneja mediante procedimientos de rutina.

2.3.3.2. Seleccionar los 10 riesgos de mayor relevancia

Se realiza una selección por parte del equipo, de los 10 riesgos que tienen mayor relevancia para el proyecto.

2.3.3.3. Dar prioridad a cada riesgo

Se desarrolla una nueva lista con los 10 riesgos más frecuentes que fueron aceptados en el paso anterior, otorgándole a cada uno su prioridad.

Se debe tener en cuenta el nivel que le fue otorgado a cada uno.

Artefacto de entrada:

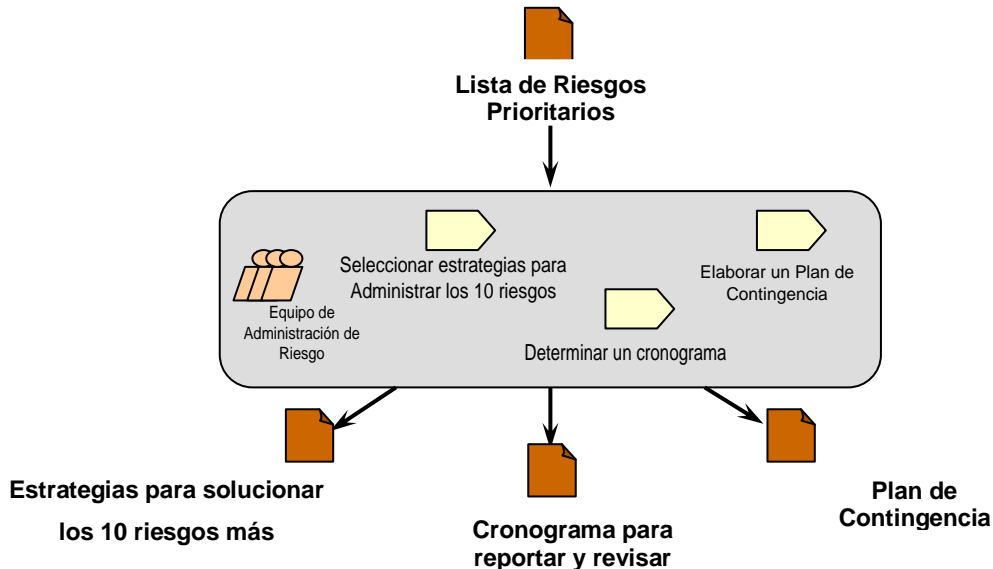
1. La lista de riesgos
2. La caracterización de los riesgos

Artefacto de salida:

1. Lista de riesgos prioritarios

2.3.4. Flujo de Trabajo: Planificación

La función de este Flujo de Trabajo es convertir la información sobre los riesgos en decisiones y acciones para el presente y el futuro.



2.3.4.1. Seleccionar estrategias para administrar los 10 riesgos

Teniendo en cuentas las siguientes estrategias, se hace una selección de aquella que sea más factible para administrar los 10 riesgos más usuales en el proyecto:

- “Evitación”: Se trata de minimizar la probabilidad de que el riesgo se presente.
- Minimización: Reducir el impacto del riesgo en el producto o en el proyecto.
- Planes de Contingencia: Este plan se pone en práctica en caso de que fallen las anteriores estrategias. Se encarga de administrar el riesgo que está vigente en ese momento.

2.3.4.2. Determinar un cronograma

Se realiza un cronograma, para el equipo de Administración de Riesgos, con el objetivo de realizar los reportes y las revisiones en tiempo y con la calidad que requieren. De esta forma, se logra acelerar el proceso para eliminarlos.

2.3.4.3. Elaborar el Plan de Contingencia

Incluye, la descripción del riesgo, los indicadores que muestran que el mismo se ha hecho realidad y las acciones a llevar a cabo, para eliminar los riesgos que están afectando el proyecto en ese momento.

Las posibles técnicas que se pueden aplicar en este paso son:

- Costo – Beneficio: Es el balance entre el costo de la implementación de las medidas preventivas y el riesgo remanente. Además tiene en cuenta la posibilidad de ocurrencia de daños materiales, tanto a la instalación, como a la pérdida de la producción durante los períodos de parada y la reparación de los daños.
- Las acciones de aversión del riesgo:
 1. Evitar el riesgo: Se elige una alternativa de menor riesgo.
 2. Controlar el riesgo: Se decide mitigar el riesgo.
 3. Asumir el riesgo: Se acepta que el riesgo ocurra.
 4. Transferir el riesgo: Se reduce el riesgo compartiéndolo.
 - 5.

Artefacto de entrada:

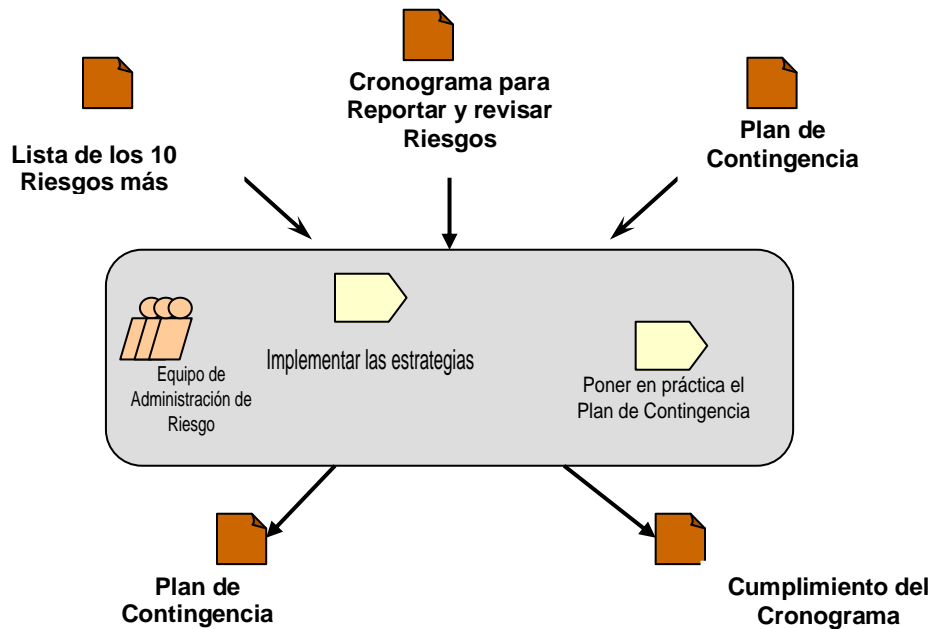
1. Lista de riesgos prioritarios.

Artefacto de salida:

1. Estrategias para solucionar los 10 riesgos más frecuentes.
2. Cronograma para reportar y revisar riesgos.
3. Plan de Contingencia.

2.3.5. Flujo de Trabajo: Resolución y/o Mitigación

Este flujo tiene la tarea de implementar las estrategias que se tomaron en el anterior flujo de trabajo y pone en práctica los planes de contingencia.



2.3.5.1. Implementar las estrategias

Se ponen en práctica las estrategias que se definieron en el flujo de trabajo de Planificación.

La mitigación define un conjunto de técnicas, que agilizan este proceso, como son:

- Poda de requisitos: Se realiza una selección de los requisitos más importantes, en caso que se tenga que realizar una primera versión del producto.
- Prototipado: Se brinda una breve panorámica acerca de las acciones que se llevarán a cabo para mitigar los riesgos.
- Desarrollo incremental: Consiste en la utilización de incrementos para aumentar gradualmente el alcance, empezando por las posibilidades más básicas de las aplicaciones y ampliándolas paso a paso.

2.3.5.2. Poner en práctica el Plan de Contingencia

Se refina y pone en práctica el plan de contingencia, elaborado en el flujo de trabajo anterior.

Artefactos de entrada:

1. Lista de los 10 riesgos más frecuentes.

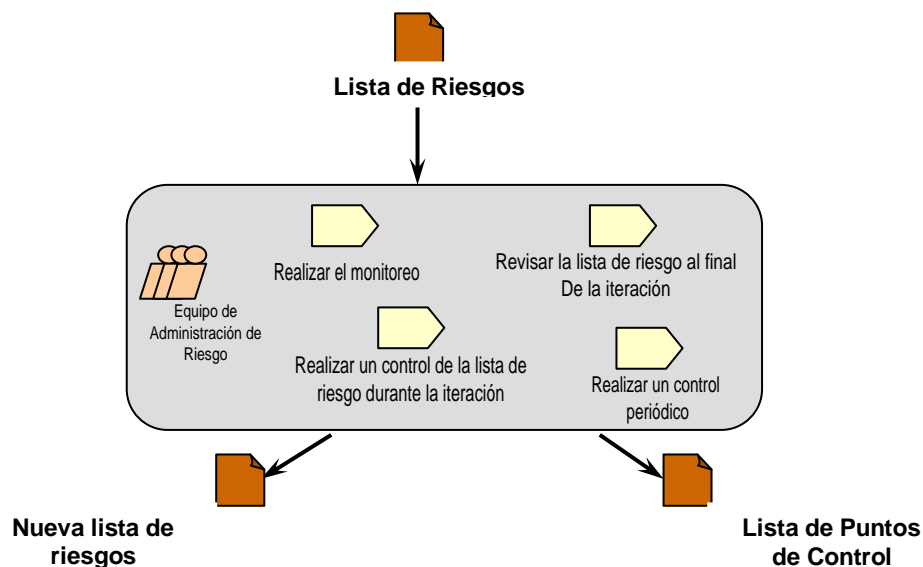
2. Cronograma para reportar y revisar riesgos.
3. Plan de Contingencia.

Artefacto de Salida:

1. Plan de contingencia aplicado.
2. Cumplimiento del cronograma.

2.3.6. Flujo de Trabajo: Monitoreo

Es una etapa esencial e integral en el proceso de Gestión de Riesgo, porque es donde se realiza un monitoreo cada cierto tiempo de los riesgos que están presentes y de los que se incluirán a medida que vaya avanzando el proyecto.



2.3.6.1. Realizar un Monitoreo

En esta actividad se realiza el monitoreo de:

1. Los riesgos.
2. La efectividad del plan.
3. Las estrategias.

2.3.6.2. Realizar un control de la lista de riesgo durante la iteración

Este paso comprende, un seguimiento de la lista de riesgos durante toda la iteración que se este desarrollando en ese momento.

2.3.6.3. Revisar la lista de riesgo al final de la iteración

Se realiza una actualización de la lista de riesgo, pero al final de la iteración.

2.3.6.4. Realizar un control periódico

El monitoreo de los riesgos debe desarrollarse periódicamente, para detectar si se han introducido nuevos riesgos.

En caso de que la detección sea efectiva, se debe pasar automáticamente al flujo de trabajo de Identificación.

Existen técnicas que se pueden aplicar durante todo el flujo de trabajo como son:

1. Revisión e inspección al proyecto.
2. Reuniones de avance de proyecto.
3. Auditorias.

Artefacto de Entrada

1. lista de riesgos

Artefacto de Salida

1. Nueva lista de riesgo.
2. Listas de punto de control.

2.4. Capacitación del Personal

Con el incremento de la actividad productiva en la UCI, hay un aumento de la necesidad de calificar al personal en los temas de Proceso de Desarrollo de Software y Gestión de Riesgo. Este último ayudará a que el personal se familiarice aún más con las formas que existen a nivel mundial.

Por eso y cumpliendo con el objetivo específico Diseñar el sistema de contenido para los temas de, Proceso de Desarrollo de Software y Gestión de Riesgo, para los estudiantes del Segundo Perfil de Calidad. A continuación se presenta el sistema de contenido:

2.4.1 Sistema de contenido de Proceso de Desarrollo de Software

Proceso de Desarrollo (Conceptos). Actividades fundamentales (Análisis (Negocio y Requerimiento), Diseño, Implementación, Validación, Verificación, Prueba, Mantenimiento). Actividades Protectores (Las planteados por Pressman). Modelos del Proceso de Desarrollo de Software (Codificar y Codificar ,Modelo en Cascada, Desarrollo Evolutivo, Desarrollo basado en Reutilización, Desarrollo Incremental, Desarrollo Iterativo, Desarrollo en Espiral). Metodologías (Ágiles y No Ágiles).

2.4.2. Sistema de contenido de Gestión de Riesgo

Riesgo (Conceptos). Estrategias (Proactiva y Reactiva).Tipos de Riesgo .Métodos generales de resolución de riesgos. Gestión (Definición).Gestión de Riesgo (Conceptos).Generaciones de la Gestión de Riesgo (Caracterizar cada una de ellas).Diferentes modelos de la gestión de riesgo .Pasos (que se realizan dentro de la Gestión de Riesgo (Identificación, Análisis, Priorización, Planificación, Resolución y/o Mitigación, Monitoreo).Caracterizar cada uno de los pasos.

2.5. Conclusiones

Se espera que la propuesta y el curso de capacitación, cumpla con las expectativas de la universidad y el Departamento central de Ingeniería y Gestión de Software. Facilitando de esta forma, llevar a cabo una buena Gestión de Riesgo en cada uno de los proyectos productivos.

CAPÍTULO 3: EVALUACIÓN DE LA PROPUESTA

Para la validación y aceptación del proceso para la Gestión de Riesgo en los proyectos productivos de la UCI, que se presenta en el Capítulo 2, se utilizó el criterio de un Grupo de Expertos. Este panel se conformó con especialistas que poseen una vasta experiencia como líderes de proyecto y en temas relacionados con este trabajo.

En esta técnica se realiza una selección del grupo de expertos que participará en el proceso de evaluación, teniendo en cuenta que ningún experto conoce la identidad y las respuestas individuales de los otros que componen el grupo. Esto posibilita, que un miembro pueda cambiar sus opiniones y defender sus argumentos con la tranquilidad de saber que en caso de ser erróneos, no representará una pérdida de su prestigio. Además, impide que un experto sea influenciado por la reputación de otro.

La correcta elección de los expertos propicia obtener resultados con calidad y una opinión grupal con un alto grado de consenso.

En el presente capítulo se hará la descripción de los pasos utilizados en la selección del Panel de Expertos y los resultados obtenidos.

3.1. Proceso de selección de los expertos

Se entiende por experto a un especialista en una materia. Persona experimentada, que posee una gran experiencia o habilidad en una actividad, capaz de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones al respecto.

En el desarrollo de este proceso se consideraron tres etapas cruciales:

3.1.1. Determinar la cantidad de expertos

Respecto al número de expertos que deben configurar el panel, no existe una norma generalizada que determine un número óptimo. Los investigadores de *Rand Corporation*, indican que es necesario como mínimo de siete expertos y un máximo de 30, según Dalkey et al, (1969) citado por (LANDETA, 1999).

En este trabajo se decidió contar con un número de ocho expertos para la confección del panel, teniendo en cuenta nivel de complejidad y profundidad del contenido.

3.1.2. Confeccionar el listado de expertos

La confección del listado de expertos se realizó atendiendo a la posibilidad real de participación de los candidatos, pues todos son profesionales de la UCI que tienen como promedio 4.5 años de experiencia en la docencia, y en el proceso productivo de la universidad. La mayoría de ellos han desempeñado, el rol de líder de proyecto en un momento determinado. Poseen además, amplios conocimientos en temas relacionados con el proceso a evaluar, estos son:

1. Calidad de Software
2. Proceso de Desarrollo de Software
3. Metodología *Rational Unified Process (RUP)*
4. Gestión de Riesgos

También se tuvo en cuenta su participación en eventos nacionales e internacionales de gran trascendencia, cursos como el de “Introducción al CMMI”, impartido por el SEI, en Ciudad de México donde dos de los expertos adquirieron su certificación.

Existe una serie de cualidades propias de estos especialistas seleccionados, que se tuvieron en cuenta por parte de las autoras de esta investigación para la confección del listado. A continuación se relacionan:

1. Seriedad
2. Honestidad
3. Sinceridad
4. Responsabilidad
5. Creatividad
6. Capacidad de análisis

Estas cualidades han permitido que las opiniones brindadas sean confiables y válidas para el objetivo propuesto. Ver Anexo V.

3.1.3. Confirmar la participación de los expertos

Una vez conformado el listado, se invitó personalmente a cada experto elegido para participar en la evaluación. Allí se les explicó en que consistía el trabajo en general, la propuesta a evaluar y el objetivo de la realización de la encuesta, así como el plazo de entrega.

Una vez recibida la respuesta positiva, se estableció el listado final de los expertos, informando a cada especialista su inclusión en el proceso a evaluar y las instrucciones necesarias para contestar las preguntas.

De esta forma culmina el proceso de selección, logrando la participación de los ocho expertos escogidos.

3.2. Elaboración del cuestionario

Para la elaboración del cuestionario se tuvieron en cuenta los objetivos, que debería cumplir el procedimiento propuesto para su implantación en los proyectos productivos de la universidad.

La encuesta establece una serie de preguntas de enfoque investigativo, que permitieron ver la posibilidad real de que pueda ser aplicada la propuesta, según las características actuales de la UCI. Además de brindar su efectividad en caso de ser establecida y una evaluación general del proceso, teniendo en cuenta una serie de requisitos y una escala que se propuso del uno al cinco, siendo uno el de menor escala y cinco el de mayor. Estas preguntas proporcionan una mayor riqueza en las respuestas que son brindadas por los especialistas. Ver Anexo VI.

Se les facilitó también, la posibilidad de modificar aspectos que ellos consideraban necesario cambiar y presentar su opinión general a favor o en contra del procedimiento propuesto, con la libertad de expresar todo lo que se pudo obviar en el cuestionario. Ver Anexo IV.

Todos los expertos recibieron a través del correo electrónico un informe con la propuesta a evaluar, el cuestionario y un plazo de tiempo determinado, para enviar sus respuestas o realizar las preguntas pertinentes, que les surgieron al estudiar el documento.

3.3. Resultados de la evaluación

Después de procesada la encuesta se determinó que todos los expertos estuvieron de acuerdo en que era de imperiosa necesidad, la posibilidad de llevar a cabo la gestión de riesgos durante el proceso de desarrollo de software, en los proyectos productivos de la UCI.

El por qué de esa respuesta, prevaleció en que la universidad no cuenta en estos momentos con un sistema que gestione los riesgos en el desarrollo de los proyectos, ni existe un antecedente a la propuesta que se quiere establecer.

No obstante, el procedimiento propuesto constituye una de las primeras intenciones para atenuar esa dificultad, a través de un conjunto de pasos imprescindibles, aprobados por todos los especialistas.

De esta manera, si se establece la propuesta en cada uno de los proyectos productivos, se pueden controlar las ejecuciones, llevar a cabo las acciones pro-activas, ayudar a la gestión del proyecto, para así lograr satisfacer las necesidades de la universidad y alcanzar mayor productividad en el desarrollo del proyecto, una mayor garantía de la Calidad en el proceso y optimización de los recursos tanto humanos como materiales.

Los expertos destacaron además, que es un proceso adaptable a los proyectos independientemente de la metodología a utilizar.

Al responder a la pregunta que indagaba sobre la efectividad de la aplicación del proceso, todos los expertos estuvieron de acuerdo en que sería positiva, porque ordenaría todo en cuanto a la manera de gestionar los riesgos en los proyectos, facilitando así el control de los mismos.

Plantearon que la problemática de no existir como precedente una propuesta similar a esta en la UCI, y la escasa cultura en cuanto al tema, pudieran tener una repercusión en contra de la correcta aplicación del procedimiento.

También señalaron como otro posible factor negativo, la falta de una mayor capacitación en los temas de Gestión de Riesgo y Proceso de Desarrollo de Software, de los equipos de trabajo de los proyectos

productivos. En este sentido los expertos consideraron de gran valor que el procedimiento que se propone, incluye un curso de capacitación en correspondencia con el Perfil docente-productivo de Calidad de Software, respondiendo a la necesidad de la puesta en práctica de este, y de la activación de dicho perfil.

3.4. Conclusiones

Con lo visto hasta el momento durante todo el proceso de validación por parte de los expertos y analizando los resultados que arrojaron sus respuestas en los cuestionarios que les fueron aplicados. Se hace imprescindible entonces, el establecimiento por parte de la universidad del procedimiento para llevar a cabo la Gestión de Riesgos en los proyectos productivos y del curso de capacitación de todo su personal.

CONCLUSIONES

- Una buena Gestión de Riesgo en todo el Proceso de Desarrollo de Software, es un tema muy tratado y por el que todos en el mundo del desarrollo de software abogan.
- Las experiencias en el proceso de Gestión de Riesgo no sólo se enmarcan en los países desarrollados, sino en todos los que se encuentren en vías de desarrollo y estén vinculados a la producción de software.
- El diagnostico analizado demuestra que el 95% de los proyectos productivos en la UCI utilizan como metodología a RUP, de allí que sea las más utilizada.
- El 53% de los encuestados afirman que sí llevan a cabo una Gestión de Riesgo durante el Proceso de Desarrollo de Software.
- La Gestión de Riesgo en los proyectos productivos de la UCI es una necesidad primordial.
- El procedimiento propuesto es efectivo, adaptable, objetivo y con amplias posibilidades de aplicación.
- La aplicación del proceso en los proyectos productivos constituye, una de las primeras intenciones para atenuar con la dificultad existente.

RECOMENDACIONES

Para el buen desempeño y obtención de los resultados esperados, se recomienda:

- No ejecutar el Proceso de Gestión de Riesgo en los proyectos productivos de la UCI, si no está claro de su objetivo y alcance; así como tampoco si el personal que lo ejecutará no está capacitado para realizarlo.
- Realizar la confección, por parte del Departamento Central de Ingeniería y Gestión de Software, del P1 y sistema de evaluación, en los dos temas: Proceso de Desarrollo de Software y Gestión de Riesgo.
- Analizar con profundidad cada una de las herramientas, que se resumen al final del Capítulo 1.
- Automatizar el Proceso de Gestión de Riesgo.

BIBLIOGRAFÍA

Citadas:

- ADUANAS, S. N. D. *El proceso de Gestión de Riesgos* [Portal del Gobierno de Chile. Servicio Nacional de Aduanas]. Chile: [Consultado el: 22 de mayo de 2007]. Disponible en: http://www.aduana.cl/p4_principal/site/artic/20050916/asocfile/ASOCFILE120050916161822.pdf.
- ASTIGARRAGA, E. *El Método DELPHI* [PDF digital]. San Sebastian: [Consultado el: 5 de abril de 2007]. Disponible en: http://www.codesyntax.com/prospectiva/Metodo_delphi.pdf.
- BUADES, G. *Calidad en Ingeniería de Software* [Sitio Web: Sesión del Curso Ingeniería del Software III, (5 ° Ingeniería Informática de la UIB, Universitat de les Illes Balears)]. Última actualización: 15 de mayo del 2003. [Consultado el: 7 de mayo de 2007]. Disponible en: <http://dmi.uib.es/~bbuades/calidad/index.htm>
- CANÓS, J.; LETELIER, P., *et al.* [Portal de la Universidad Politécnica de Valencia]. Valencia. España: [Consultado el: 30 de marzo de 2007]. Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- CONNELL, S. *Capítulo 5: Desarrollo y Gestión de Proyectos Informáticos*. Traducido por: Development, R. McGraw-Hill, 1997.
- CONSULTORES, A. *La Calidad Total* [Portal de Aiteco Consultores. Calidad en la Atención al Cliente. Gestión de Procesos Sociedad inscrita en el Registro Mercantil de Granada. Tomo 863, Libro0, Folio 180, Hoja: GR 15 189]. Granada. España: Última actualización: 2006. [Consultado el: 16 de mayo de 2007]. Disponible en: http://www.aiteco.com/calidad/index.php?option=com_content&task=view&id=18&Itemid=27.
- CONTE, P. *Secretos de la gestión de proyectos de software* [Portal Web: HELP400]. Publicaciones HELP400, Última actualización: 12 de junio del 2007. [Consultado el: 5 de mayo de 2007]. Disponible en: <http://www.help400.es/asp/scripts/nwart.asp?Num=171&Pag=28&Tip=M>.
- DIRECCIÓN DE LA CARRERA DE INFORMÁTICA DE LA UTEZ, U. T. E. Z. *Calidad en el Desarrollo de Software. Guía del Alumno* México: [Consultado el: 12 de marzo de 2007]. Disponible en: http://fenix.utim.edu.mx/man/general/biblioteca/info2004/Calidad_Desarrollo_Software.pdf.
- ELCHE, U. M. H. D. *Concepto de Calidad* [Curso de Calidad. Área de Gestión y Control de la Calidad en el Portal de la Universidad Miguel Hernández de Elche]. Alicante: Última actualización: 2007.

[Consultado el: 12 de febrero de 2007]. Disponible en:
<http://calidad.umh.es/curso/documentos/concepto.pdf>

ESTÉVEZ, M. L. B. y GALLASTEGUI, J. J. A. *El Método Delphi. Su implementación en una Estrategia Didáctica para la Enseñanza de las Demostraciones Geométricas*. publicado el: 3 de marzo del 2007 de 2002, última actualización: 3 de marzo del 2007. 11 p. Disponible en:
<http://www.rieoei.org/deloslectores/804Bravo.PDF>. ISBN 1681-5653.

FAJARDO, C. *Gestión De Riesgo* [Portal de la Universidad de Vigo. Escuela Superior de Ingeniería Informática. Planificación de Proyectos Informáticos]. Última actualización: 2006. [Consultado el: 22 de mayo de 2007]. Disponible en:
http://trevinca.ei.uvigo.es/~cfajardo/Nueva_carpeta/presentaciones/PPI-t4_3.ppt.

FIGUEROA, E. G. *Calidad Total y Políticas de Recursos Humanos en el Sector Hotelero de Cataluña*. Tutor: Icart, D. I. B. Doctoral, Facultad de Ciencias Económicas y Empresariales. Departamento de Gestión de Empresas. Universitat Rovira i Virgili, 2004.

FUENTE, A. A. J. y LOVELLE, J. M. C. *Gestión de Riesgo* [PDF digital]. [Consultado el: 19 de enero de 2007]. Disponible en:
<http://www.di.uniovi.es/~aquilino/Asignaturas/ProyectosInformatica/Documentos/Proyectos.v2006.C7.V2.pdf>.

FUTRELL, S. *Quality Software Project Managment*. Editorial Prentice Hall, 2002. ISBN 0130912972.

GATTACA, I. P. *Presentación de Metodología MSF (Microsoft Solutions Framework)* [Documento digital]. Última actualización: 2006. [Consultado el: 14 de abril de 2007]. Disponible en:
<http://www.e-gattaca.com/eContent/library/documents/DocNewsNo50DocumentNo6.PDF>.

GIL, M. *Editorial*. 1 ed. 2006, vol. 1, 120 p. ISBN Solicitado.

LANDETA, J. *El método Delphi: Una técnica de previsión para la incertidumbre*. 1999,

MARECOS, D. E. A. *Revista de Posgrado de la Cátedra Vía Medicina*. publicado el: 3 de marzo del 2007 de 2001, última actualización: 3 de marzo del 2007. Disponible en:
http://med.unne.edu.ar/revista/revista108/con_claves_salud.htm.

MEDINA, D. E. F. *Ciclo de Vida del Software* [Sitio Web del Grupo Alarcos (formado por profesores del Departamento de Tecnologías y Sistemas de Información de la Universidad de Castilla-La Mancha). Ingeniería del Software I. Curso 2006-2007]. Ciudad Real. España: [Consultado el: 21 de marzo de 2007]. Disponible en: <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema03.pdf>

- MENDOZA, M. A. *Metodologías De Desarrollo De Software* [Sitio Web: Informatizate]. Perú: Última actualización: 2002. [Consultado el: 14 de abril de 2007]. Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.
- MURCIA, U. D. *Gestión de riesgos en ingeniería del software* [Portal de la Universidad de Murcia]. Última actualización: 30 de diciembre del 2006. [Consultado el: 15 de febrero de 2007]. Disponible en: <http://www.um.es/docencia/barzana/IAGP/Iagp5.html>.
- NAVARRO, A. *Gestión de Riesgo* [PDF digital]. [Consultado el: 16 de abril de 2007]. Disponible en: http://209.85.165.104/search!=cache:9k0ediu_2b0j:www.fld.ucm.es/profesor/jlsierra/lsl/.
- ORANTES, S. D. Calidad de Software en el uso de Metodologías Agiles para el Desarrollo de Software. En *Tercer Taller de Calidad en las Tecnologías de la Información y las Comunicaciones. Informática 2007* Palacio de Convenciones de la Habana-Cuba 2007. p. 25.
- ORANTES, S. D. y CASTILLO, A. B. Una Generación de Servicios Web con Calidad. En *Tercer Taller de Calidad en las Tecnologías de la Información y las Comunicaciones. Informática 2007. Palacio de Convenciones de la Habana-Cuba. Febrero 2007*. p. 16.
- PMI. *Capítulo 11: A Guide to the Project Management Body of Knowledge*. Estados Unidos de America: 2000.
- PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico*. Vedado, Ciudad de La Habana, Cuba: 2005. vol. 1, 601 p.
- REYNOX. *Metodologías de Desarrollo de Software* [Sitio Web: REYNOX. Servicios Informáticos]. Argentina: Última actualización: 2005. [Consultado el: 5 de abril de 2007]. Disponible en: <http://www.reynox.com.ar/sap/metodologia.php>.
- ROPPONEN, J. y LYYTINEN, K. *Components of Software Development Risk: How to address Them?* *IEEE transactions on software engineering*. 2000. vol. 26,
- ROSENBERG, L.; HAMMER, T., *et al. Continuous Risk Management*. En San José - California. 1999.
- RUBIO, G. B. *Ingeniería de Software III* [Portal de la Universidad de Islas Baleares]. Última actualización: 15 de mayo del 2003. [Consultado el: 10 de febrero de 2007]. Disponible en: <http://dmi.uib.es/~bbuades/riesgos/index.htm>.
- RUP, A. E. D. [Ayuda del Rational]. 2003,

- SÁNCHEZ, I. L. F. *Gestión de Riesgo en la fase de Ingeniería de Requisitos de un proyecto de software* Las Villas.Cuba: Última actualización: 2006. [Consultado el: 20 de enero de 2007]. Disponible en: <http://www.monografias.com/trabajos41/riesgo-etapa-requisitos/riesgo-etapa-requisitos2.shtml>.
- SEI. *Continuous Risk Managment Guidebook*. Carnegie University: 2004.
- SEVILLA, U. D. *Gestion de Riesgos* [Portal de la Universidad de Sevilla]. Sevilla. España: Última actualización: 2006. [Consultado el: 12 de marzo de 2007]. Disponible en: <http://www.lsi.us.es/docencia/get.php?id=1784>.
- VALENCIA, U. P. D. *Proceso de Desarrollo de Software* [Portal de la Universidad Politécnica de Valencia. Departamento de Sistemas Informáticos y Computación]. Valencia. España: [Consultado el: 12 de marzo de 2007]. Disponible en: <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20Proceso%20de%20Desarrollo%20de%20SW.doc>.
- ZAVALA, J. *Ingeniería de Software. Extracto de una tesis de maestría* [Sitio Web: Angelfire. Sitio personal de Jesus Zavala]. Última actualización: 8 de septiembre del 2002. [Consultado el: 16 de enero de 2007]. Disponible en: <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>,

Consultada:

- ADR, F. *La Calidad Total* [Plataforma de formación, dedicada a la formación y al desarrollo de aplicaciones informaticas. Cursos de enseñanza Online]. Logroño: Última actualización: 2005. [Consultado el: 4 de abril de 2007]. Disponible en: <http://www.adrformacion.com/cursos/calidad/leccion1/tutorial4.html>
- AERONÁUTICOS, E. T. S. D. I. *Concepto de Calidad* [Sitio web: Unidad de Producción Aeroespacial de la Escuela Técnica Superior de Ingenieros Aeronáuticos de la Universidad Politécnica de Madrid]. Madrid: Última actualización: 13 de abril del 2005. [Consultado el: 12 de febrero de 2007]. Disponible en: http://fabetsia.dmpa.upm.es/solo_alumnos/sp2/Tablon_sp2/TransparenciasCALIDAD06.pdf
- ALCÓN, S.; FORNASARI, A., *et al.* El concepto de calidad total aplicada a un Centro de Información. En *Cuarto Encuentro para la Integración de Bibliotecas y Centros de Documentación de las Fuerzas Armadas*. Buenos Aires. Argentina. 1993.
- ALQUICIRA, C. *Modelo de Procesos para la Industria del Software (MoProSoft)* [Sitio Web de la Universidad Católica de San Pedro]. Última actualización: 24 de febrero del 2007 [Consultado el: 19 de mayo de 2007]. Disponible en:

http://inf.ucsp.edu.pe/summerschool/Formal%20Method%20in%20Software%20Development_files/MoproSoft/MoProsoft%20Escuela%20Verano%20Peru.ppt

- CARRASCO, O. F.; LEÓN, D. G., *et al.* *Un enfoque actual sobre la Calidad de Software*. publicado el: 20 de mayo del 2007 de 1995, última actualización: 20 de mayo del 2007. vol. 3, 1 p. Disponible en: http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm. ISBN 1530:2880
- CASTRO, R. A. *Estructura básica del proceso unificado de desarrollo de software* [Trabajo publicado de la Universidad Icesi]. [Consultado el: 20 de abril de 2007]. Disponible en: <http://cursweb.educadis.uson.mx/aalba/documentos/RUP.pdf>
- CONSULTORES, Á. D. C. D. I. *Calidad Total-EFQM-ISO 9000 ¿Diferencias y Similitudes?* publicado el: 3 de junio del 2007 de 2001, última actualización: 3 de junio del 2007. 4 p. Disponible en: http://www.improven.com/Pdf/ISO_EFQM.pdf
- DANIELE, M.; ROMERO, D., *et al.* La Enseñanza de Gestión de Proyectos de Software y la aplicación de herramientas que favorezcan su automatización. En *Tercer Taller de Calidad en las Tecnologías de la Información y las Comunicaciones. Informática 2007. Palacio de Convenciones de la Habana-Cuba. 2007.* p. 10.
- DELGADO, M. D. L. C. y MARCOS, F. S. *Proceso de Desarrollo con UML y el modelo CMM* [Presentación digital]. [Consultado el: 20 de abril de 2007]. Disponible en: <http://www.ati.es/gt/calidad-software/SIMO00/SIMO2000-UML.ppt#256>
- FILHO, C. V. F. y REIS, J. M. D. *Herramienta de apoyo al proceso de desarrollo de software en las pequeñas compañías.* 2006, 7 p. Disponible en: <http://www.inf.udec.cl/~revista/ediciones/edicion12/articulo%2012-6.pdf>
- GONZÁLEZ, C. *Concepto de calidad total en diseño.* publicado el: 12 de marzo del 2007 de 2005, última actualización: 12 de marzo del 2007. 1 p. Disponible en: http://www.raulybarra.com/notijoya/archivosnotijoya4/4economia_calidad_total.htm
- GONZÁLEZ, C. D. *Desarrollo de Software. Software de aplicación. Software de línea de productos. Aplicaciones basadas en Web* [Sitio Web: usabilidadweb.com.ar Consultoría en Usabilidad y Estándares web]. Buenos Aires. Argentina: Última actualización: 2007. [Consultado el: 16 de mayo de 2007]. Disponible en: http://www.usabilidadweb.com.ar/ingenieria_software.php
- IMPROVEN-CONSULTORES. *Entendiendo el concepto de Calidad* [Portal TodoMBA, buscador, ranking, noticias y becas de Masters]. Todo MBA, S.L. ed. España: Última actualización: 2006. [Consultado el: 3 de junio de 2007]. Disponible en: http://www.todomba.com/index2.php?option=com_content&do_pdf=1&id=1126
- JENKINS, M. Comparación de las iniciativas latinoamericanas para mejorar la Industria del Software. En *Tercer Taller de Calidad en las Tecnologías de la Información y las Comunicaciones. Informática*

2007. *Palacio de Convenciones de la Habana-Cuba 2007*. p. 10.

JIMÉNEZ, M. H. F. A. *Modelos de ciclo de vida en desarrollo de software*. Publicado el: 13 de mayo del 2007 de 2005, última actualización: 13 de mayo del 2007. 1 p. Disponible en: <http://www.acis.org.co/index.php?id=551>

LOVELLE, J. M. C. Calidad de Software. En *Grupo de GIDIS. Universidad Nacional de la Pampa. 21 de octubre 1999*. p. 12.

MONTAÑO, R. M. *Portal de la Universidad Tecnológica Emiliano Zapata del Estado de Morelos* [Sitio Web. Portal de la universidad]. Estado Morelos, México: [Consultado el: 13 de junio de 2007]. Disponible en: <http://www.utez.edu.mx/index.php>

PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico*. Quinta ed. Madrid: 2002.

ROMÁN, I. R.; CARREIRA, M. R., et al. *Los Modelos Dinámicos y la Ingeniería del Software* [Documento digital]. [Consultado el: 5 de abril de 2007]. Disponible en: <http://www.sc.ehu.es/jiwdocoj/remis/docs/modelos.html>

ROSIQUE, G. N. *Presentación de monografía sobre Calidad del Software / Software de Calidad*. Publicado el: 8 de junio del 2007 de 1999, última actualización: 8 de junio del 2007. 1 p. Disponible en: <http://www.ati.es/novatica/1999/137/pres137.html>

TORO, V. M. *Gerencia de Proyectos <=>Proceso de Software* [PDF digital de la Compañía de Ingenieros Constructores de Software]. Bogotá. Colombia: [Consultado el: 30 de marzo de 2007]. Disponible en: <http://www.cincosoft.com>

VÁZQUEZ, R. H. *Taller de Calidad de Software, Introducción a la Calidad del Software* [Sitio Web de la Universidad Tecnológica Nacional. Departamento de Sistemas de Información de la Facultad Regional Mendoza]. [Consultado el: 8 de junio de 2007]. Disponible en: <http://web.frm.utn.edu.ar/liredat/docs/Introduccion%20a%20la%20Calidad%20de%20Software%20Vazquez.pdf>

ANEXO I

Listas de Comprobación (Checklists)

Se construyen a partir de información histórica.

Se va a emplear una lista muy completa, que agrupa los riesgos de proyectos software en las siguientes categorías:

- A. Elaboración de la Planificación
- B. Organización y Gestión
- C. Ambiente/Infraestructura de Desarrollo
- D. Usuarios finales
- E. Cliente
- F. Personal Contratado
- G. Requisitos
- H. Producto
- I. Fuerzas Mayores
- J. Personal
- K. Diseño e Implementación
- L. Proceso

A. Elaboración de la Planificación

- A.1. Las definiciones de la planificación, de los recursos y del producto han sido impuestas por el cliente o un directivo superior, y no están equilibradas.
- A.2. Planificación optimista, «mejor caso» (en lugar de realista, «caso esperado»).
- A.3. La planificación no incluye tareas necesarias.
- A.4. La planificación se ha basado en la utilización de personas específicas de un equipo, pero estas personas no están disponibles.
- A.5. No se puede construir un producto de tal envergadura en el tiempo asignado.
- A.6. El producto es más grande que el estimado (en líneas de código, en el número de puntos función, o en relación con el tamaño del proyecto anterior).
- A.7. El esfuerzo es mayor que el estimado (por líneas de código, número de puntos función, módulos, etc.).
- A.8. La reestimación debida a un retraso en la planificación es demasiado optimista o ignora la historia del proyecto.
- A.9. La presión excesiva en la planificación reduce la productividad.
- A.10. La fecha final ha cambiado sin ajustarse al ámbito del producto o a los recursos disponibles.
- A.11. Un retraso en una tarea produce retrasos en cascada en las tareas dependientes.
- A.12. Las áreas desconocidas del producto llevan más tiempo del esperado en el diseño y en la implementación.

B. Organización y Gestión

- B.1. El proyecto carece de un promotor efectivo en los superiores.
- B.2. El proyecto languidece demasiado en el inicio difuso.

- B.3. Los despidos y las reducciones de la plantilla reducen la capacidad del equipo.
- B.4. Dirección o marketing insisten en tomar decisiones técnicas que alargan la planificación.
- B.5. La estructura inadecuada de un equipo reduce la productividad.
- B.6. El ciclo de revisión/decisión de la directiva es más lento de lo esperado.
- B.7. El presupuesto varía el plan del proyecto.
- B.8. La dirección toma decisiones que reducen la motivación del equipo de desarrollo.
- B.9. Las tareas no técnicas encargadas a terceros necesitan más tiempo del esperado (aprobación del presupuesto, aprobación de la adquisición de material, revisiones legales, seguridad, etc.).
- B.10. La planificación es demasiado mala para ajustarse a la velocidad de desarrollo deseada.
- B.11. Los planes del proyecto se abandonan por la presión, llevando al caos y a un desarrollo ineficiente.
- B.12. La dirección pone más énfasis en las heroicidades que en informarse exactamente del estado, lo que reduce su habilidad para detectar y corregir problemas.

C. Ambiente/Infraestructura de Desarrollo

- C.1. Los espacios no están disponibles en el momento necesario.
- C.2. Los espacios están disponibles pero no son adecuados (por ejemplo, falta de teléfonos, cableado de la red, mobiliario, material de oficina, etc.).
- C.3. Los espacios están sobre utilizados, son ruidosos o distraen.
- C.4. Las herramientas de desarrollo no están disponibles en el momento deseado.
- C.5. Las herramientas de desarrollo no funcionan como se esperaba; el personal de desarrollo necesita tiempo para resolverlo o adaptarse a las nuevas herramientas.
- C.6. Las herramientas de desarrollo no se han elegido en función de sus características técnicas, y no proporcionan las prestaciones previstas.
- C.7. La curva de aprendizaje para la nueva herramienta de desarrollo es más larga de lo esperado.

D. Usuarios Finales

- D.1. Los usuarios finales insisten en nuevos requisitos.
- D.2. En el último momento, a los usuarios finales no les gusta el producto, por lo que hay que volver a diseñarlo y a construirlo.
- D.3. Los usuarios no han realizado la compra del material necesario para el proyecto y, por tanto, no tienen la infraestructura necesaria.
- D.4. No se ha solicitado información al usuario, por lo que el producto al final no se ajusta a las necesidades del usuario, y hay que volver a crear el producto.

E. Cliente

- E.1. El cliente insiste en nuevos requisitos.
- E.2. Los ciclos de revisión/decisión del cliente para los planes, prototipos y especificaciones son más lentos de lo esperado.
- E.3. El cliente no participa en los ciclos de revisión de los planes, prototipos y especificaciones, o es incapaz de hacerlo, resultando unos requisitos inestables y la necesidad de realizar unos cambios que consumen tiempo.

- E.4. El tiempo de comunicación del cliente (por ejemplo, tiempo para responder a las preguntas para aclarar los requisitos) es más lento del esperado.
- E.5. El cliente insiste en las decisiones técnicas' que alargan la planificación.
- E.6. El cliente intenta controlar el proceso de desarrollo, con lo que el progreso es más lento de lo esperado.
- E.7. Los componentes suministrados por el cliente no son adecuados para el producto que se está desarrollando, por lo que se tiene que hacer un trabajo extra de diseño e integración.
- E.8. Los componentes suministrados por el cliente tienen poca calidad, por lo que tienen que hacerse trabajos extra de comprobación, diseño e integración.
- E.9. Las herramientas de soporte y entornos impuestos por el cliente son incompatibles, tienen un bajo rendimiento o no funcionan de forma adecuada, con lo que se reduce la productividad.
- E.10. El cliente no acepta el software entregado, incluso aunque cumpla todas sus especificaciones.
- E.11. El cliente piensa en una velocidad de desarrollo que el personal de desarrollo no puede alcanzar.

F. Personal Contratado

- F.1. El personal contratado no suministra los componentes en el período establecido.
- F.2. El personal contratado proporciona material de una calidad inaceptable, por lo que hay que añadir un tiempo extra para mejorar la calidad.
- F.3. Los proveedores no se integran en el proyecto, con lo que no se alcanza el nivel de rendimiento que se necesita.

G. Requisitos

- G.1. Los requisitos se han adaptado, pero continúan cambiando.
- G.2. Los requisitos no se han definido correctamente y su redefinición aumenta el ámbito del proyecto.
- G.3. Se añaden requisitos extra.
- G.4. Las partes del proyecto que se no se han especificado claramente consumen más tiempo del esperado.

H. Producto

- H.1. Los módulos propensos a tener errores necesitan más trabajo de comprobación, diseño e implementación.
- H.2. Una calidad no aceptable requiere de un trabajo de comprobación, diseño e implementación superior al esperado.
- H.3. Utilizar lo último en informática alarga la planificación de forma impredecible.
- H.4. El desarrollo de funciones software erróneas requiere volver a diseñarlas y a implementarlas.
- H.5. El desarrollo de una interfaz de usuario inadecuada requiere volver a diseñarla y a implementarla.
- H.6. El desarrollo de funciones software innecesario alarga la planificación.
- H.7. Alcanzar el ámbito del producto o las restricciones de velocidad requiere más tiempo del esperado, incluyendo el tiempo para volver a diseñar e implementar.
- H.8. Unos requisitos rígidos de compatibilidad con el sistema existente necesitan un trabajo extra de comprobación, diseño e implementación.

- H.9. Los requisitos para crear interfaces con otros sistemas, otros sistemas complejos, u otros sistemas que no están bajo el control del equipo de desarrollo suponen un diseño, implementación y prueba no previstos.
- H.10.El requisito de trabajar con varios sistemas operativos necesita más tiempo del esperado.
- H.11.El trabajo con un entorno software desconocido causa problemas no previstos.
- H.12.El trabajo con un entorno hardware desconocido causa problemas imprevistos.
- H.13.El desarrollo de un tipo de componente nuevo para la organización consume más tiempo del esperado.
- H.14.Depender de una tecnología que aún está en fase de desarrollo alarga la planificación.

I. Fuerzas mayores

- I.1. El producto depende de las normativas del gobierno, que pueden cambiar de forma inesperada.
- I.2. El producto depende de estándares técnicos provisionales, que pueden cambiar de forma inesperada.

J. Personal

- J.1. La contratación tarda más de lo esperado.
- J.2. Las tareas preliminares (por ejemplo, formación, finalización de otros proyectos, adquisición de licencias) no se han completado a tiempo.
- J.3. La falta de relaciones entre la dirección y el equipo de desarrollo ralentiza la toma de decisiones.
- J.4. Los miembros del equipo no se implican en el proyecto, y por lo tanto no alcanzan el nivel de rendimiento deseado.
- J.5. La falta de motivación y de moral reduce la productividad.
- J.6. La falta de la especialización necesaria aumenta los defectos y la necesidad de repetir el trabajo.
- J.7. El personal necesita un tiempo extra para acostumbrarse a trabajar con herramientas o entornos nuevos.
- J.8. El personal necesita un tiempo extra para acostumbrarse a trabajar con hardware nuevo.
- J.9. El personal necesita un tiempo extra para aprender un lenguaje de programación nuevo.
- J.10. El personal contratado abandona el proyecto antes de su finalización.
- J.11. Alguien de la plantilla abandona el proyecto antes de su finalización.
- J.12. La incorporación de nuevo personal de desarrollo al proyecto ya avanzado, y el aprendizaje y comunicaciones extra imprevistas reducen la eficiencia de los miembros del equipo existentes.
- J.13. Los miembros del equipo no trabajan bien juntos.
- J.14. Los conflictos entre los miembros del equipo conducen a problemas en la comunicación y en el diseño, errores en la interfaz y tener que repetir algunos trabajos.
- J.15. Los miembros problemáticos de un equipo no son apartados, influyendo negativamente en la motivación del resto del equipo.
- J.16. Las personas más apropiadas para trabajar en el proyecto no están disponibles.
- J.17. Las personas más apropiadas para trabajar en el proyecto están disponibles, pero no se pueden incorporar por razones políticas o de otro tipo.
- J.18. Se necesitan personas para el proyecto con habilidades muy específicas y no se encuentran.
- J.19. Las personas clave sólo están disponibles una parte del tiempo.
- J.20. No hay suficiente personal disponible para el proyecto.
- J.21. Las tareas asignadas al personal no se ajustan a sus posibilidades.

- J.22. El personal trabaja más lento de lo esperado.
- J.23. El sabotaje por parte de la dirección del proyecto deriva en una planificación ineficiente e inefectiva.
- J.24. El sabotaje por parte del personal técnico deriva en una pérdida de trabajo o en un trabajo de poca calidad, por lo que hay que repetir algunos trabajos.

K. Diseño e Implementación

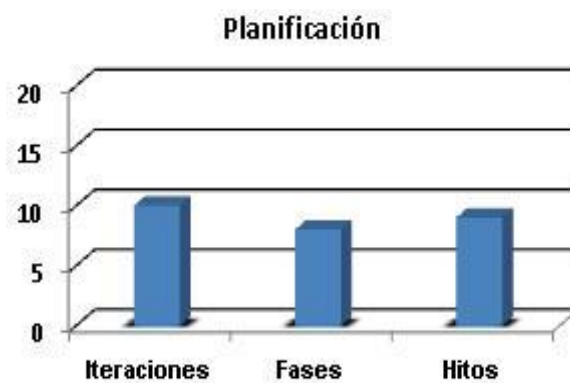
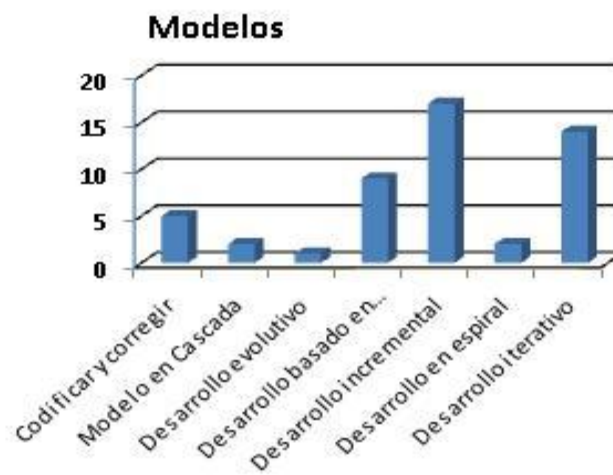
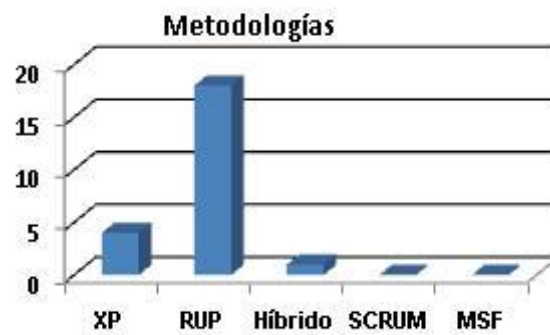
- K.1. Un diseño demasiado sencillo no cubre las cuestiones principales, con lo que hay que volver a diseñar e implementar.
- K.2. Un diseño demasiado complejo exige tener en cuenta complicaciones innecesarias e improductivas en la implementación.
- K.3. Un mal diseño implica volver a diseñar e implementar.
- K.4. La utilización de metodologías desconocidas deriva en un periodo extra de formación y tener que volver atrás para corregir los errores iniciales cometidos en la metodología.
- K.5. El producto está implementado en un lenguaje de bajo nivel (por ejemplo, ensamblador) y la productividad es menor de la esperada.
- K.6. No se puede implementar la funcionalidad deseada con el lenguaje o bibliotecas utilizados: el personal de desarrollo tiene que utilizar otras bibliotecas, o crearlas él mismo para conseguir la funcionalidad deseada.
- K.7. Las bibliotecas de código o clases tienen poca calidad, y generan una comprobación extra, corrección de errores y la repetición de algunos trabajos.
- K.8. Se ha sobreestimado el ahorro en la planificación derivado del uso de herramientas para mejorar la productividad.
- K.9. Los componentes desarrollados por separado no se pueden integrar de forma sencilla, teniendo que volver a diseñar y repetir algunos trabajos.

L. Proceso

- L.1. La burocracia produce un progreso más lento del esperado.
- L.2. La falta de un seguimiento exacto del progreso hace que se desconozca que el proyecto esté retrasado hasta que está muy avanzado.
- L.3. Las actividades iniciales de control de calidad son recortadas, haciendo que se tenga que repetir el trabajo.
- L.4. Un control de calidad inadecuado hace que los problemas de calidad que afectan a la planificación se conozcan tarde.
- L.5. La falta de rigor (ignorar los fundamentos y estándares del desarrollo de software) conduce a fallos de comunicación, problemas de calidad y repetición del trabajo. Un consumo de tiempo innecesario.
- L.6. El exceso de rigor (aferramiento burocrático a las políticas y estándares de software) lleva a gastar más tiempo en gestión del necesario.
- L.7. La creación de informes de estado a nivel de directiva lleva más tiempo al desarrollador de lo esperado.
- L.8. La falta de entusiasmo en la gestión de riesgos impide detectar los riesgos más importantes del proyecto.
- L.9. La gestión de riesgos del proyecto software consume más tiempo del esperado.

ANEXO II

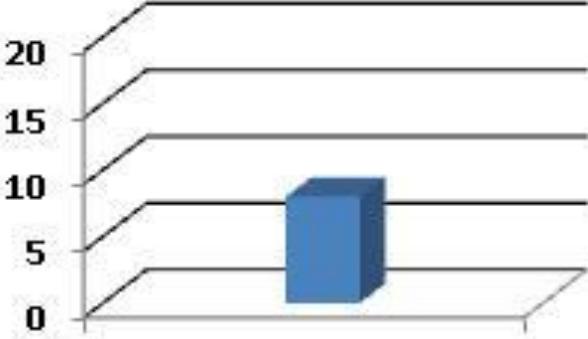
Proceso de Desarrollo de Software



Gestión de Riesgo



Monitoreo



ANEXO III

Encuesta en los Proyectos Productivos.**Proceso de Desarrollo de Software.**

- 1) De las siguientes actividades fundamentales que se encuentran presentes durante el proceso de desarrollo, cuál de ellas ustedes aplican.
 - Análisis (Negocio y requerimiento).
 - Diseño e Implementación.
 - Validación, verificación y prueba.
 - Evolución.

- 2) Pressman menciona un conjunto de actividades protectoras, cual de ellas se aplica en su proyecto.
 - Seguimiento y control de proyecto de software.
 - Revisiones técnicas formales.
 - Garantía de la Calidad de Software.
 - Gestión de Configuración del Software.
 - Preparación y producción de documentos.
 - Gestión de reutilización.
 - Mediciones.
 - Gestión de Riesgos.
 - Prueba.
 - Evaluación del producto.
 - Auditorias.

- 3) ¿Qué modelos del proceso de software aplican?
 - Codificar y corregir.
 - Modelo en Cascada.
 - Desarrollo evolutivo.
 - Desarrollo basado en reutilización.
 - Desarrollo incremental.
 - Desarrollo en espiral.

- 4) ¿Qué metodología aplican para llevar a cabo el desarrollo del software?

Gestión de Riesgo.

1) ¿Ustedes llevan a cabo una Gestión de Riesgo?

Si No

2) ¿Qué estrategia utilizan para la Gestión de Riesgo?

Proactiva Reactiva

3) Diga 5 riesgos que atacan frecuentemente a su proyecto.

4) En caso de realizar la Gestión de Riesgo:

¿Qué modelo utilizan?

Modelo de Boehm.

Modelo de Gestión de Riesgo en Base a Taxonomías.

Modelo de Magerit.

5) Según el modelo que utiliza diga si tiene en cuenta algunos de estos pasos

Identificación de riesgo.

Análisis de riesgo.

Priorización de riesgo.

Planeación de riesgo.

Resolución de riesgo.

Monitorización de riesgo.

ANEXO IV

Encuesta realizada a expertos

Questionario

- 1) ¿Cree usted que el procedimiento propuesto está a la altura de las posibilidades y necesidades de la UCI?

Sí___ No___ ¿Por qué?

- 2) Con la propuesta establecida en los proyectos, ¿cree usted lograr efectividad en el manejo de los riesgos?

Sí___ No___ ¿Por qué?

- 3) ¿Considera usted que los pasos aludidos en el procedimiento propuesto son los necesarios para realizar la Gestión de Riesgo?

Sí___ No___ ¿Por qué?

Si cree preciso proponer o eliminar alguno, méncionelo y explique brevemente.

- 4) En la escala del 1 al 5 otorgue una evaluación al procedimiento propuesto según los criterios siguientes:

___ Satisfacción de las necesidades de la UCI.

___ Adaptabilidad a los proyectos productivos independientemente de la metodología a utilizar.

___ Repercusión en los proyectos productivos.

___ Posibilidad de aplicación.

- 5) ¿Cuáles serían los argumentos que usted expondría a favor de la aplicación del procedimiento propuesto y cuáles estarían en contra?

- 6) Realice alguna observación o aporte sobre la propuesta que es por su parte objeto de su evaluación.

ANEXO V

Currículo de Expertos.

Expertos	Graduado de	Años Vinculados A la UCI	Curso CMMI	Eventos Científicos	Experiencia	Otros Cursos
Experto 1	Ing. Informática, 2000	4	Si	Convenciones de Informática. Sepgla 2004. Metánica de calidad. Uciencia	En pruebas de Software: <ul style="list-style-type: none"> • Registros y Notarías • Identidad • Digitalización • Multimedias • Intranet de PDVSA • Pruebas de aceptación en el extranjero Líder de Proyecto de Calidad	
Experto 2	Ing. Informática, 2003	4	No	Convenciones de Informática. Sepgla 2004 y 2005 Sisoft 2005 Uciencias	Líder de Proyecto de Identidad	
Experto 3	Ing Sistemas Automatizados (1992), Maestría en Informática Aplicada, 1995	5	Si	Convenciones Informáticas	Jefa del Departamento de Ing y Gestión de Software	
Experto 4	Ing. Informática, 2005	2	No	Uciencias 2006	Líder de Proyecto Lims Control de Calidad del CIGB	Dos Cursos de Gestión de Proyectos

Experto 5	Ing. Informática, 2005	2	No	Convenciones Informáticas Uciencia 2005 y 2006 Congreso de Reconocimien tos de Patrones 2005	Líder de Proyecto con el Centro Nacional de Genética Médica Líder del Modulo de Registro y Notarias	Curso de Gestión de Proyectos
Experto 6	Ing. Informática, 2005	2	No	Convenciones Informáticas Uciencia 2005 y 2006 Congreso de Reconocimien tos de Patrones 2005	Líder de Proyecto de Bioinformática y Registro y Notaria	Cursos de la Maestría Diplomado de Docencia Universitaria Gestión de Proyectos
Experto 7	Ing. Cibernética - Matemática, 1996	5	No	Convenciones Informáticas	Líder de Proyecto	Developer Java Applications Administra ción de Oracle 10g Programaci ón en PL\SQL
Experto 8	Ing. Cibernética, 1990	5	No	Convenciones Informáticas Uciencia 2005 y 2006	Líder de Proyecto	

ANEXO VI

Criterio de expertos

Aspectos a evaluar	Exp.1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6	Exp.7	Exp.8
Satisfacción de las necesidades de la UCI	5	4	4	5	5	4	4	5
Adaptabilidad en los proyectos productivos independiente de la metodología a utilizar	5	1	4	4	5	4	3	4
Repercusión en los proyectos productivos	5	5	4	4	5	5	5	4
Posibilidad de aplicación	5	4	5	5	4	5	5	5

GLOSARIO

Boehm: Gran matemático, el cual inventó COCOMO.

CMMI (*Capability Maturity Model Integration*): es un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Además, se desarrolló para facilitar y simplificar la adopción de varios modelos de forma simultánea, y su contenido integra y da relevancia a la evolución de sus predecesores. Fue desarrollado por el Instituto de Ingeniería del Software de la Universidad Carnegie Mellon (SEI), y publicado en su primera versión en enero de 2002.

IEEE: Es el Instituto de Ingenieros Eléctricos y Electrónicos es una Asociación Profesional técnica sin fines de lucro de más de 350,000 miembros individuales en 175 países. el IEEE es la principal autoridad en áreas técnicas que van de la Ingeniería en Computación, Ingeniería Eléctrica, Aeroespacial, Electrónica, Tecnología Biomédica, Telecomunicaciones, Electrónica de consumo, entre otros.

Esta información fue encontrada en una página Web llamada Sitio Web Oficial de IEEE. Inc. Sección Nicaragua.

Inexorable: Viene del latín “inexorabilis”, negativo de “exorabilis”, derivado del verbo latino “exorable”, que significa “rogar”, y a su vez también lo es de “orar, rogar, suplicar”. Inexorable es algo que se cumplirá porque el que lo ha decidido no se deja ablandar. (Este significado fue tomado de una página Web llamada “HISPANOTECA”, lenguaje y cultura)

ISO (*International Organization for Standardization* o *Organización Internacional para la Estandarización*): es una organización internacional no gubernamental, compuesta por tres tipos de miembros, uno miembros natos, miembros correspondientes y por último miembros suscritos, además, de los representantes de los organismos de normalización nacionales que produce normas internacionales industriales y comerciales. Dichas normas se conocen como normas ISO, que tienen como propósito facilitar el comercio, el intercambio de información y contribuir con unos estándares comunes para el desarrollo y transferencia de tecnologías.

Roger. S. Pressman: Es una autoridad internacionalmente reconocida en la mejora de proceso de software y en tecnología de Ingeniería de Software. Por más de tres décadas, ha trabajado como ingeniero, gerente, profesor, autor y consultor de software en temas de Ingeniería de Software. Actualmente es presidente de R. S. Pressman and Asóciate, Inc., una firma consultora especialista en métodos y entrenamiento en Ingeniería de Software. Es conocido también como EL PADRE DE LA INGENIRIA DE SOFTWARE.

RUP: El *Rational Unified Process (RUP)* es una metodología formal, a veces también llamada proceso. El RUP describe a gran detalle todas las actividades, roles, responsabilidades, productos de trabajo y herramientas para definir quién hace qué y en qué momento en un proyecto de desarrollo de software.

SEI: Instituto de Ingeniería de Software es un organismo financiado por el gobierno federal de los Estados Unidos y operado por la universidad Carnegie Mellon de ese país. El SEI existe para ayudar a las organizaciones a mejorar el estado de las prácticas de ingeniería, con el fin de incrementar la calidad de los sistemas que dependen del software y la ingeniería de sistemas.

Taxonomía: Las Taxonomías son las clasificaciones ordenadas de elementos de acuerdo a sus relaciones presumidas; y pueden emplearse como herramientas de suma utilidad en diferentes ramas de la ciencia y la industria donde se pretende organizar y facilitar el acceso a un número importante de elementos que se encuentran mutuamente relacionados de alguna manera relevante.