

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA DE SISTEMAS**

**PROPUESTA DE APLICACIÓN DE SCRUM PARA MINIMIZAR LOS RIESGOS EN UN PROYECTO DE DESARROLLO DE SOFTWARE.**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**LILIÁN ELIZABETH ARROBA MEDINA**

**`lilian_elizabeth22@hotmail.com`**

**DIRECTOR: MSC. ING. JAIME FABIÁN NARANJO ANDA**

**`jaime.naranjo@epn.edu.ec`**

**Quito, Abril 2011**

## **DECLARACIÓN**

Yo Lilián Elizabeth Arroba Medina, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

**Lilián Elizabeth Arroba Medina**

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue desarrollado por Lilián Elizabeth Arroba Medina bajo mi supervisión.

---

**Msc. Ing. Jaime Naranjo**  
**DIRECTOR DE PROYECTO**

## **AGRADECIMIENTOS**

A mi familia que me apoyado en cada etapa de mi vida, a Dios por brindarme fuerza y sabiduría, a mis amigas y amigos con los cuales he compartido momentos buenos y malos en estos últimos años, convirtiéndose en una parte muy importante de mi vida,

Al Ing. Jaime Naranjo, que con sus valiosos consejos no solo han aportado a la realización de la presente tesis, sino que me han ayudado a mi crecimiento personal y espiritual.

## **DEDICATORIA**

A mis padres por haberme brindado oportunidades valiosas de superación en mi vida, por haber hecho de cada buen momento, un logro compartido.

A mis hermanas y hermano que siempre han sido un apoyo en cada momento de mi vida, convirtiéndose en ejemplos de virtudes.

A mis sobrinos y sobrinas, de los cuales me siento muy orgullosa, porque a pesar del tiempo conservan esa inocencia de niños que mi familia les ha sabido inculcar.

# CONTENIDO

CAPÍTULO I.....	1
DIMENSIONES EN LOS PROYECTOS DE.....	1
DESARROLLO DE SOFTWARE.....	1
1.1. COMPARACIÓN DE METODOLOGÍAS ÁGILES Y TRADICIONALES.....	1
1.2. VISIÓN ACTUAL DE LAS METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE.....	7
1.2.1. PORQUE USAR METODOLOGÍAS ÁGILES.....	7
1.2.2. EL MANIFIESTO ÁGIL.....	12
CAPÍTULO II.....	17
ANÁLISIS DE SCRUM.....	17
2.1. CARACTERÍSTICAS DE SCRUM.....	17
2.1.1. CARACTERÍSTICAS.....	18
2.1.2. ELEMENTOS DE SCRUM.....	20
2.1.2.1. Roles.....	20
2.1.2.2. Documentos requeridos en Scrum.....	21
2.1.2.3. Reuniones.....	23
2.2. HERRAMIENTAS UTILIZADAS EN SCRUM.....	26
2.2.1. GRÁFICO BURN-UP.....	26
2.2.2. GRÁFICO BURN-DOWN.....	27
2.2.3. JUEGO O PROTOCOLO DE DECISIÓN - ESTIMACIÓN DE PÓKER.....	28
2.2.4. DESARROLLO DE RETROSPECTIVA.....	30
2.2.5. DESARROLLO DE DOCUMENTO PRODUCT BACKLOG.....	31
2.2.6. DESARROLLO DE DOCUMENTO SPRINT BACKLOG.....	32
2.3. DESCRIPCIÓN DEL FUNCIONAMIENTO Y BENEFICIOS DE SCRUM.....	35
2.3.1. FUNCIONAMIENTO DE SCRUM.....	35
2.3.2. BENEFICIOS DE SCRUM.....	39
2.3.3. LIMITACIONES DE SCRUM.....	40

CAPÍTULO III.....	42
DESARROLLO DE PROPUESTA PARA LOGRAR.....	42
REDUCCIÓN DE RIESGOS EN PROYECTOS.....	42
DE SOFTWARE UTILIZANDO SCRUM. ....	42
3.1. RIESGOS COMUNES EN PROYECTOS ACTUALES DE PROYECTOS DE SOFTWARE.....	42
3.2. PROPUESTA PARA LOGRAR LA REDUCCIÓN DE RIESGOS.....	49
3.2.1. CONSIDERACIONES DE LA PROPUESTA.....	49
3.2.2. OBJETIVOS DE LA PROPUESTA. ....	50
3.2.3. PROPUESTA. ....	51
CAPÍTULO IV.....	68
APLICACIÓN DE LA PROPUESTA UTILIZANDO SCRUM.....	68
4.1. CASO DE ESTUDIO.....	68
4.1.1. DESCRIPCIÓN DE LA EMPRESA.....	68
4.1.2. DESCRIPCIÓN DEL SISTEMA. ....	68
4.1.3. OBJETIVOS DE LA APLICACIÓN DE LA PROPUESTA.....	69
4.2. APLICACIÓN DE LA PROPUESTA Y ANÁLISIS DE RESULTADOS. ....	70
4.2.1. APLICACIÓN DE LA PROPUESTA.....	70
4.2.2. ANÁLISIS DE RESULTADOS. ....	88
CAPITULO V.....	90
CONCLUSIONES Y RECOMENDACIONES.....	90
5.1. CONCLUSIONES.....	90
5.2. RECOMENDACIONES.....	93
REFERENCIAS BIBLIOGRAFICAS. ....	95
REFERENCIAS EN INTERNET.....	95
ANEXOS. ....	98
ANEXO A.....	98
Glosario de términos en inglés.....	98
ANEXO B.....	98
Carta de la empresa – conclusiones sobre aplicación de la propuesta. ....	98

## LISTA DE FIGURAS

Figura 1. Gráfico Burn-Up.....	27
Figura 2. Gráfica Burn-Down. ....	28
Figura 3. Juego o Protocolo de decisión – Estimación de Póker.....	30
Figura 4. Representación de Formato de Product Backlog – Pizarrón.....	34
Figura 5. Metodología Scrum – Proceso. ....	35
Figura 6. Ciclo de gestión de riesgos. ....	45
Figura 7. Proceso de gestión de riesgos, basado en la realización de objetivos.....	46
Figura 8. Representación del avance de gestión de riesgos en un pizarrón. ....	63

## LISTA DE TABLAS

Tabla 1. Diferencias entre metodologías tradicionales y ágiles.....	7
Tabla 2. Comparación de metodologías ágiles.....	14
Tabla 3. Convergencias y divergencias entre las principales metodologías ágiles. ....	15
Tabla 4. Formato de Product Backlog. ....	32
Tabla 5. Representación de Formato de Product Backlog – Hola electrónica.....	34
Tabla 6. Categoría versus componentes.....	48
Tabla 7. Resumen de etapas de la propuesta para la reducción de riesgos. ....	51

## LISTA DE FORMATOS

Formato 1. Documento de identificación de riesgos genéricos. ....	53
Formato 2. Documento de identificación de riesgos específicos.....	55
Formato 3. Documento para priorización de riesgos. ....	58
Formato 4. Documento para gestión de riesgos.....	60
Formato 5. Documento para control de solución de riesgos.....	62
Formato 6. Documento para análisis de gestión de riesgos.....	65
Formato 7. Documento para retrospectiva de riesgos.....	67

## LISTA DE DOCUMENTOS.

Documento 1. Identificación de riesgos genéricos- Iteración 1.....	71
Documento 2. Identificación de riesgos específicos- Iteración 1. ....	72
Documento 3. Priorización de riesgos - Iteración 1. ....	73
Documento 4. Gestión de riesgos - Iteración 1.....	74
Documento 5. Control de solución de riesgos - Iteración 1. ....	75
Documento 6. Análisis de gestión de riesgos - Iteración 1-2-3.....	76
Documento 7. Identificación de riesgos genéricos- Iteración 2.....	77
Documento 8. Identificación de riesgos específicos- Iteración 2. ....	78
Documento 9. Priorización de riesgos - Iteración 2. ....	79
Documento 10. Gestión de riesgos - Iteración 2.....	80
Documento 11. Control de solución de riesgos - Iteración 2. ....	81
Documento 12. Identificación de riesgos genéricos- Iteración 3.....	83
Documento 13. Identificación de riesgos específicos- Iteración 3. ....	84
Documento 14. Priorización de riesgos - Iteración 3. ....	85
Documento 15. Gestión de riesgos - Iteración 3.....	86
Documento 16. Control de solución de riesgos - Iteración 3. ....	87

## RESUMEN

En la Ingeniería de Software actualmente los proyectos de desarrollo de software se encuentran respaldados por diferentes tipos de metodologías, las mismas que brindan una guía y base de conocimientos, que permiten realizar este tipo de proyectos de mejor manera, entre estas metodologías se encuentran las nominadas ágiles, que brindan una proyección más amplia y flexible en el desarrollo de sistemas.

Una de las metodologías ágiles es Scrum, que permite realizar un proceso lo suficientemente manejable y adaptable a los proyectos de desarrollo de software, facilitando el acoplamiento de todos los elementos involucrados en los mismos. Esta metodología brinda grandes beneficios a los proyectos, debido a su apertura ante posibles cambios que puedan sufrir en el transcurso del desarrollo, tomando estos como posibilidades de mejora continua, para asegurar que el software futuro sea de calidad y brinde beneficios al negocio, los mismos que llegan de forma temprana en las primeras iteraciones.

Los proyectos de desarrollo de software en el transcurso de la historia han sufrido inconvenientes, los mismos que han tratado de ser solventados, de manera que no afecten drásticamente al sistema en desarrollo, y así reducir el impacto negativo sobre este, es así que se vio factible la posibilidad de crear una propuesta para reducir los riesgos en proyectos de desarrollo de software, basada en el proceso que plantea la metodología Scrum, ya que esta propone la gestión de riesgos implícitamente durante todo el ciclo de vida de los proyectos.

## **PRESENTACIÓN**

El presente proyecto de titulación, está encaminado a la creación de una propuesta para reducir los riesgos en proyectos de desarrollo de software, basada tanto en las características como en el proceso planteado por la metodología Scrum. Esta propuesta está dirigida a encaminar y guiar al grupo de trabajo con la ejecución de una gestión correcta y oportuna de riesgos, durante el desarrollo del proyecto de software.

En los dos primeros capítulos de la tesis, se revisan conceptos sobre metodologías ágiles y sobre el proceso que plantea Scrum para desarrollar software; todos estos conceptos ayudan y soportan la estructuración de la propuesta planteada en el capítulo 3.

En la parte final se realiza la ejecución práctica de la propuesta a un proyecto de desarrollo de software, logrando una mejor gestión de riesgos por parte de los integrantes del grupo de desarrollo.

En base a la información desarrollada en el presente proyecto de titulación, finalmente se plantearon conclusiones y recomendaciones.

# **CAPÍTULO I.**

## **DIMENSIONES EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE.**

### **1.1. COMPARACIÓN DE METODOLOGÍAS ÁGILES Y TRADICIONALES.**

En el mundo actual una de las metas principales de las industrias es reducir los trabajos manuales y transformarlos en procesos automáticos por medio de sistemas de información. El desarrollo de estos sistemas en muchas ocasiones toma un tiempo prolongado por lo que incurre en pérdidas económicas para las industrias, ya que en ocasiones no se tiene lo suficientemente claro lo que se desea y se tiene disponible, como: funcionalidad, tiempo, y presupuesto; de esta manera no se tiene entendido el proceso a seguir, originando etapas en el desarrollo que se combinan sin ningún análisis previo, es ahí que en medio de muchas confusiones, el uso de metodologías para el desarrollo de software ayuda en gran medida en tener un control durante el proceso.

El uso de las diferentes metodologías de desarrollo de software, es una decisión muy importante para alcanzar la calidad durante los procesos que implican la actividad de una empresa, esto va acompañado de diferentes pasos, los que comprenden un control sistemático y beneficioso para el negocio, tratando de esta manera satisfacer todas las necesidades del usuario.

En el mercado actual existen diferentes tipos de metodologías, entre las que se puede nombrar las conocidas como tradicionales y las ágiles, a continuación se nombrarán las diferencias existentes entre ellas, en la Tabla 1:

<b>Enfoque de la metodología.</b>	
<p><b>Tradicional:</b></p> <p>El enfoque en el que se basan estas metodologías es el desarrollo en cascada, es decir, que cada etapa del proceso de desarrollo se ejecutará una a continuación de otra, en un periodo determinado de tiempo, y solo en este espacio se realizará dicha fase, por lo que no existe flexibilidad para retroceder y realizar alguna revisión, de ser el caso en fases anteriores a la que se ejecuta en ese momento.</p>	<p><b>Ágil:</b></p> <p>El desarrollo por medio de iteraciones es la base fundamental de este tipo de metodologías, en cada una de estas se realizan todas las fases del proceso de desarrollo, entregando así al cliente un producto altamente usable en cada iteración, y sobre el cual se puedan realizar observaciones para la respectiva retroalimentación del siguiente entregable.</p>
<b>Grupo de Trabajo.</b>	
<p><b>Tradicional:</b></p> <p>En este tipo de metodologías los roles del grupo de trabajo se manejan de forma muy estricta, sin poder realizar mayor cambio de ser necesario; además el recurso humano no es tomado como un eje principal para el proceso de desarrollo, tornándose según su filosofía, en algo fácilmente reemplazable.</p>	<p><b>Ágil:</b></p> <p>La multifuncionalidad y la auto organización que caracterizan a los miembros del grupo de trabajo que desarrollan con este tipo de metodologías, permite que se puedan realizar cambios de ser necesario en los roles de grupo de trabajo. Estas metodologías recalcan la importancia del recurso humano, y del aporte que brinda cada miembro del grupo al proyecto, fomentando la colaboración constante.</p>
<b>Planificación.</b>	
<p><b>Tradicional:</b></p> <p>La planificación se establece en la etapa inicial en base a todas las fases que</p>	<p><b>Ágil:</b></p> <p>Si bien en este tipo de metodologías también se establece una planificación de</p>

<p>implican el proyecto de desarrollo, la cual se proyecta que se cumplirá de manera muy precisa para no afectar la calidad del producto final. En este proceso de programación, el grupo de trabajo suele dedicar gran esfuerzo para su desarrollo, con esto se intenta predecir con exactitud el desarrollo del proyecto, calculando para esto, tiempo y recursos que procuren ser lo más correctos posible.</p>	<p>trabajo previa, esta debe ser cumplida a medida de lo posible, ya que el cambio y modificación de fechas, recursos o alcance, no afecte al éxito del proyecto. Estas metodologías no enfocan la funcionalidad del sistema en la planificación previa, sino en la adaptación a cambios que puede presentarse durante el desarrollo del proyecto.</p>
<b>Flexibilidad para cambios.</b>	
<p><b>Tradicional:</b> Las metodologías tradicionales debido a la filosofía de cumplimiento estricto de la planificación realizada previamente, no brindan mayor apertura a realizar cambios en el proyecto, ya que estos afectan a la predictibilidad que se intenta manejar en el desarrollo.</p>	<p><b>Ágil:</b> Las metodologías ágiles tienen un alto nivel de adaptabilidad a los cambios, tomando a estos no como riesgos altos en la mayoría de casos, sino como posibilidades para retroalimentar el proyecto y obtener el mayor beneficio.</p>
<b>Requerimientos.</b>	
<p><b>Tradicional:</b> Los requerimientos se tratan en este tipo de metodologías se procura manejarlos de forma estática, es decir, que si inicialmente se fijo un requerimiento este no cambie en el transcurso del proyecto, ya que sobre este ya se realizó una planificación previa, y la modificación de</p>	<p><b>Ágil:</b> Los requerimientos en la etapa inicial tratan de ser entendidos de manera global y muy claramente, las posibles variaciones son tomadas como mejoras, que agregarán un valor significativo al producto final.</p>

este, sería tomado como un riesgo alto para el proyecto.	
<b>Riesgos.</b>	
<p><b>Tradicional:</b></p> <p>Los riesgos para proyectos que usan estas metodologías incrementan, debido a que al tratar de manejar la mayor predictibilidad posible, los cambios que se puedan presentar en el transcurso del proyecto hace que las metas impuestas no se cumplan, y dado que en la mayoría de proyectos existen cambios esto hace que exista incoherencia tratando de usar metodologías que se desean cero variación en su planeación, en un ambiente del negocio que tiene un alto grado de transformación.</p>	<p><b>Ágil:</b></p> <p>Debido al enfoque de adaptabilidad a cambios de estas metodologías, permite que en cada iteración se acelere la reducción de riesgos, ya que la retroalimentación ayuda a que posibles problemas que se presenten en las primeras iteraciones, puedan desaparecer a medida que el proyecto avanza.</p>
<b>Comunicación.</b>	
<p><b>Tradicional:</b></p> <p>La comunicación dependerá del nivel de interacción que el líder del grupo tenga con el resto de los integrantes, como se citó anteriormente el recurso humano no toma la verdadera importancia para estas metodologías, y la capacidad de decisión es muy limitada, con excepción del líder. Si bien si existe un proceso de intercambio de ideas y opiniones, pero no es visto como</p>	<p><b>Ágil:</b></p> <p>La comunicación es imprescindible en este tipo de metodologías, con el constante intercambio de información se logrará la retroalimentación deseada, siendo así un elemento clave para el éxito del proyecto, esta debe ser en todo sentido, es decir, tanto entre los miembros del grupo de desarrollo como con el cliente. Se debe recordar que el factor</p>

una necesidad hacerlo con frecuencia.	humano es un eje principal de este tipo de metodologías.
<b>Documentación.</b>	
<p><b>Tradicional:</b></p> <p>La documentación exhaustiva que se genera con este tipo de metodologías, en ocasiones crea un sentimiento en el grupo de trabajo que este ha desperdiciando esfuerzo, en algo que tal vez no aporta gran valor al proyecto, pero es importante recalcar que hay organizaciones para las cuales mientras más documentación exista es mucho mejor, con esto se cree que no existirá dependencia por ningún miembro del grupo, y este podrá ser fácilmente reemplazable, ya que tendrá una fuerte base de conocimiento en papel.</p>	<p><b>Ágil:</b></p> <p>La generación de documentación se realizará solo en el caso de ser necesario, para que de esta manera el esfuerzo del grupo sea utilizado en otras actividades del proyecto que se creen que aportarán mayor valor al proyecto. La creación de documentación deberá ser manejada por el criterio de grupo, basado en experiencias previas de trabajo, generando así la información imprescindible que ayude a la toma de decisiones.</p>
<b>Cliente.</b>	
<p><b>Tradicional:</b></p> <p>El cliente es tomado como un recurso de cierta forma externo al grupo de trabajo, con el cual se acuerda los requerimientos en la etapa inicial del proyecto, a partir del cual se realizará la negociación contractual, y al cual se le presentará el producto final, que en el mejor de los casos con el tiempo y recursos definidos en la etapa inicial.</p>	<p><b>Ágil:</b></p> <p>El cliente es un miembro más del grupo de trabajo con el cual se mantiene una constante comunicación, para poder conocer las necesidades del negocio en todo momento, y de esta manera crear en el cliente un sentimiento de compromiso, y responsabilidad para lograr el éxito del proyecto.</p>

<b>Control.</b>	
<p><b>Tradicional:</b></p> <p>El control se ejerce sobre los miembros del grupo de una forma muy rigurosa y estricta, lo que no permite la apropiada interacción para lograr acuerdos, por lo que el grupo de trabajo tal vez sienta la continua imposición del líder, limitando la capacidad de opinión.</p>	<p><b>Ágil:</b></p> <p>Debido a la constante comunicación entre los miembros del grupo, solamente es oportuno realizar un control sutil, ya que al existir un intercambio constante de información, se puede detectar a tiempo cualquier inconveniente, y poder encontrar la solución óptima con el conocimiento que posee el grupo, con el poder de decisión adecuado.</p>
<b>Contrato.</b>	
<p><b>Tradicional:</b></p> <p>Este tipo de metodologías buscan cumplir siempre con lo planificado inicialmente, por lo que se plantea un contrato con un valor fijo y un tiempo ya preestablecido, tomando en cuenta solamente los imprevistos comunes de cualquier proyecto, y no aquellos que implicarían cambios drásticos en la planificación, con lo que los parámetros que se fijaron inicialmente en el contrato variarían, y estos ya no podrían brindar los réditos económicos esperados para el grupo de desarrollo.</p>	<p><b>Ágil:</b></p> <p>En estas metodologías se promueve la flexibilidad, por lo que se debe indicar y acordar con el cliente que el contrato no podrá tener parámetros fijos, pero si se podrá realizar una visión previa de este, para que el cliente tenga un panorama y una idea del valor y tiempo que se empleará en el proyecto, pero es muy importante recalcar que esto se podrá modificar a medida que el proyecto avance, y las necesidades del negocio ameriten cambios.</p>

<b>Orientación.</b>	
<p><b>Tradicional:</b></p> <p>Estas metodologías son orientadas a los procesos, es decir se centran en el cumplimiento estricto del ciclo de vida de un proyecto de desarrollo de software en forma de cascada.</p>	<p><b>Ágil:</b></p> <p>Las metodologías ágiles se orientan hacia las personas, se trata de lograr que las personas entiendan la importancia de su trabajo, y forman parte activa del proyecto, aportando con valor significativo y constante.</p>
<b>Forma de Trabajo.</b>	
<p><b>Tradicional:</b></p> <p>Las metodologías tradicionales buscan ser predictivas en la mayoría del tiempo de vida del proyecto, por tal razón es de suma importancia que la planificación hecha en la etapa inicial del proyecto se cumpla en un 100%.</p>	<p><b>Ágil:</b></p> <p>En estas metodologías se maneja la adaptabilidad a los cambios que se puedan presentar durante el proyecto, de esta manera se procura provocar la menor cantidad de problemas que puedan afectar al proyecto.</p>

Tabla 1. Diferencias entre metodologías tradicionales y ágiles.

Autor: Lilián Elizabeth Arroba Medina.

## **1.2. VISIÓN ACTUAL DE LAS METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE.**

### **1.2.1. PORQUE USAR METODOLOGÍAS ÁGILES.**

“En el año 2001 en Snowbird Utah, adoptaron el término metodologías ágiles, lo cual condujo a que se formara la Alianza Ágil, una organización sin fines de lucro que promueve el desarrollo ágil de aplicaciones.”(Desarrollo ágil de software [2])

Algunas de las metodologías ágiles son las siguientes: (Desarrollo ágil de software [2])

- Adaptive Software Development (ASD).
- Agile Unified Process (AUP).
- Crystal Clear.
- Essential Unified Process (EssUP).
- Feature Driven Development (FDD).
- Lean Software Development (LSD).
- Open Unified Process (OpenUP).
- Programación Extrema (XP).
- Scrum.

Se ha visto en el transcurso del tiempo que las diferentes metodologías han tenido un gran alcance en los proyectos de desarrollo de software, destacando que en las últimas décadas las metodologías ágiles han surgido como una opción muy viable, ya que estas permiten una mayor adaptabilidad a cambios. Estas metodologías aparecen en contraste a las que manejan métodos muy estructurados y estrictos, que brindan un grado de flexibilidad muy reducido a proyectos de desarrollo de software; haciendo de esta manera que las metodologías ágiles tengan gran acogida actualmente, además es uno de los temas más recientes en la ingeniería de software, logrando que el interés en estas generen una fuerte proyección industrial, combinando perfectamente los requisitos variables y nuevas tecnologías.

La ingeniería de software a diferencia de otras ingenierías, maneja objetos y conceptos intangibles, por lo que la mayoría de usuarios finales tienen la idea que los componentes de un software son fácilmente modificables, si bien esto en muchos casos es verdad, pero existen metodologías que por brindar un modelo de desarrollo muy estricto, no permiten ninguna variación en los componentes de sistemas que se encuentran en proceso de desarrollo, sin embargo existen otro tipo de metodologías llamadas ágiles, que toman las modificaciones como una evolución hacia el futuro software final esto se ve reflejado en los proyectos de software, en los cuales en su

tiempo de vida manejan tres variables: tiempo, presupuesto y calidad, en la filosofía de las metodologías tradicionales las dos primeras variables efectuadas estrictamente según lo planificado, logran la tercera; sin embargo es muy complicado que la planificación se ejecute de manera exacta, ya que en la mayoría de casos sufre algunas variaciones; en contraste con las metodologías ágiles, las cuales manejan las mismas variables, sin embargo, la modificación de las mismas no significa necesariamente un problema para el proyecto, para esto se debe contar también con la colaboración del cliente, quien debe tener claro al inicio del proyecto que el tiempo, el presupuesto y el alcance pueden variar, según las necesidades propias del negocio.

En un mundo tan imprevisible como el de hoy es difícil tratar de manejar para el desarrollo de software metodologías previsibles o tradicionales, las cuales no se encuentran listas para cambios no planeados. Una opción de manejar la reducida previsibilidad que se obtiene actualmente con las metodologías tradicionales, es contar con una constante retroalimentación, si esto se transfiere al ambiente de desarrollo de proyectos de software, la clave sería realizar el proceso por medio de iteraciones, esto es lo que brindan las metodologías ágiles. En proyectos desarrollados con estas metodologías, se va presentando los requerimientos cubiertos con la prioridad acordada con el cliente, a medida que se desarrollan las iteraciones; enfatizando en cada una la retroalimentación, lo que permite tomar los puntos de beneficio para el proyecto, y desechar aquellos que afectan o no ayudan al éxito del mismo; es decir, la retroalimentación e incorporación de nuevas ideas es imprescindible, ya que el entorno de los proyectos es cambiante, y las metodologías ágiles resultan de gran apoyo, de esta manera no arriesgan la calidad del producto final, además logran que el tamaño del proceso de desarrollo sea el necesario para lograr que el esfuerzo realizado valga la pena.

“Las metodologías ágiles brindan un medio de control de la imprevisibilidad que es la adaptabilidad” (Los Métodos Ágiles [12]). Es difícil realizar planes para un proyecto de desarrollo de software en el que se espera que se cumpla de manera muy precisa, la

mayoría de proyectos duran como mínimo pocos meses y en algunos casos años, por lo que en este tiempo es normal pensar que existirán factores que podrán modificar lo ya previsto inicialmente, con las metodologías ágiles esta planificación se realiza con cada iteración, ya que en cada ciclo del proyecto se puede ir acoplando las variantes del entorno que vayan surgiendo a medida que el desarrollo avance, siendo así un ejemplo los productos cuales se sacan al mercado como versiones mejoradas de las anteriores, en periodos cortos de tiempo, entre cada una, estos han brindando a las industrias actuales mayor prestigio, ya que se van adaptando a las necesidades actuales, que avanzan a un ritmo muy acelerado, de esta manera cada nuevo producto logrará un mayor nivel de competitividad, todo esto es respaldado por proyectos desarrollados con metodologías ágiles, que no solo permiten manejar proyectos de corta duración, sino pueden adaptarse ha aquellos que toman periodos largos de desarrollo.

Una de las características que hace actualmente que el uso de metodologías ágiles sea más común en diferentes tipo de proyectos, es que estas se centran en el factor humano, tanto el equipo de desarrollo como el cliente; además de que las diferentes iteraciones permiten ir desarrollando la constante comunicación, la que debe existir en todo sentido entre los miembros del proyecto. De esta forma, siendo uno de los ejes primordiales el recurso humano en el desarrollo del proyecto, es lo que también ayudará a determinar el éxito o el fracaso del mismo. Hay que tomar en cuenta que no solo es indispensable que cada individuo posea conocimientos, habilidades y virtudes que las pueda manejar de manera individual, sino que además deben tener la capacidad de poder interactuar con el grupo de trabajo y crear un ambiente lo suficientemente estable y cordial, lo que ayudará a que la mejora continua, promovida por las metodologías ágiles, lo que se convertirá en una costumbre en el grupo de trabajo, ayudando a generar retroalimentación, que aporte valor para el proyecto, como para el grupo, y además para cada miembro de manera individual.

Los seres humanos no son entes predecibles, esto va a la par con lo que manejan las metodologías ágiles, ya que estas operan de mejor manera con entornos altamente variantes, lo que involucra a todo tipo de recursos involucrados en el proyecto, que pueden cambiar con las circunstancias, además de esto dichas metodologías respaldan que el recurso humano no es fácilmente reemplazable, enfatizando que el costo de cambiar un recurso puede ser alto, y este a la vez no puede traer beneficios al proyecto, porque se deberá trabajar con alguien nuevo que prácticamente parte desde cero.

Para las metodologías ágiles el cliente es tomado como un recurso más del grupo de trabajo, lo que ayuda de manera impresionante en el intercambio de información, esto ayuda a que el cliente no se visualiza a sí mismo como alguien externo sino como alguien que afecta el desarrollo del producto significativamente. Dado que el cliente es quien conoce el negocio, por lo tanto podrá priorizar los requerimientos que son indispensables para la organización, logrando que mientras el sistema se vaya completando con todos los requerimientos durante todas las iteraciones, pueda ir ya representando beneficios económicos para la organización, en las primera presentaciones o iteraciones, lo que también puede significar la reducción de gastos para el negocio, a diferencia de las metodologías tradicionales que gastan muchos recursos y presupuesto en las primeras fases de desarrollo, sin brindar algún producto previo que ya le pueda generar ingresos extras a la empresa, sino solamente hasta el final del proyecto. Las metodologías ágiles brindan flexibilidad, para que estas se adapten a la organización y no viceversa, lo que podrá agregar valor a la empresa con la aplicación de prácticas útiles. Al manejar el desarrollo por medio de iteraciones se intenta reducir los tiempos de desarrollo, lo que de ninguna forma afectará la calidad del producto, ya que no se debe olvidar que en cada iteración se entrega un subproducto que ya es usable y de calidad.

Las metodologías ágiles al enfocarse en el desarrollo incremental, brindan una forma conveniente de manejar entornos que puedan ser mejor controlados en cada iteración, procurando identificar los riesgos de manera oportuna, a la vez que al manejar

intervalos cortos entre cada presentación las personas pueden imponerse las responsabilidades que creen que pueden cumplir en el tiempo seleccionado, lo que permite un trabajo más eficiente y de mejor calidad, porque cada miembro tiene el conocimiento y la autoridad de poder tomar decisiones que afecten tanto a ellos como al grupo, adquiriendo auto motivación.

### 1.2.2. EL MANIFIESTO ÁGIL

En el manifiesto ágil se valoran los siguientes puntos:

- **Al individuo, las interacciones del equipo de desarrollo, el proceso y las herramientas:** dado que el recurso humano es uno de los principales factores para el éxito del proyecto, es de suma importancia lograr que el equipo de trabajo cree su propio entorno, y no que este se adapte a uno ya prefabricado, el mismo que no podría cumplir precisamente con todos sus necesidades.
- **Desarrollar software de calidad más que conseguir una buena documentación:** se deben producir únicamente los documentos que sean realmente necesarios para respaldar la decisión en el proyecto con la información fundamental.
- **La colaboración del cliente más que la negociación de un contrato:** es de vital importancia que exista una constante comunicación con el cliente, de forma que el sienta que es un miembro más del grupo de trabajo y no solamente quien en muchas ocasiones impone las reglas, procurando con esto garantizar el éxito del proyecto.
- **Responde a los cambios más que seguir estrictamente un plan:** es claro que se debe realizar una planificación previa pero esta debe ser lo suficientemente flexible para adaptarse al entorno que podría variar drásticamente.

En base a los puntos anteriores se generan los siguientes doce principios del manifiesto ágil:( Metodologías Ágiles en el Desarrollo de Software [15])

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Se debe dar la bienvenida a los cambios, los cuales podrán ayudar para que el cliente tenga una ventaja competitiva.
- Es importante entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- La construcción del proyecto se debe realizar en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- El software que funciona es la medida principal de progreso.
- Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

A continuación en la Tabla 2 se detalla ciertas características de algunas metodologías ágiles, asignando un número a los distintos parámetros en comparación, siendo 5 el número máximo y 1 el mínimo. Se observa que la mayoría de metodologías obtienen un buen resultado en las prácticas ágiles seleccionadas.

	<b>ASD</b>	<b>Crystal</b>	<b>FDD</b>	<b>Scrum</b>	<b>XP</b>
<b>Sistema como algo cambiante</b>	5	4	3	5	5
<b>Colaboración</b>	5	5	4	5	5
<b>Resultados</b>	5	5	4	5	5
<b>Simplicidad</b>	4	4	5	5	5
<b>Adaptabilidad</b>	5	5	3	4	3
<b>Excelencia técnica</b>	3	3	4	3	4
<b>Prácticas de colaboración</b>	5	5	3	5	5
<b>Media Total</b>	4.5	4.4	3.7	4.5	4.5

Tabla 2. Comparación de metodologías ágiles.

Fuente: Metodologías ágiles para el desarrollo de software:  
eXtreme Programming (XP) [16]

En la Tabla 3 se describen algunas metodologías ágiles con sus características como: acrónimo, desarrollador y fecha, tipo de modelo, descripción.

Metodología	Acrónimo	Desarrollador/ Fecha de Creación	Tipo de modelo	Característica
Adaptive Software Development	ASD	Highsmith / 2000	Prácticas + ciclo de vida	Inspirado en sistemas adaptativos complejos
Agile Modeling	AM	Ambler / 2002	Metodología basada en la práctica	Suministra modelado ágil a otros métodos
Cristal Methods	CM	Cockbum / 1998	Familia de metodologías	Metodología ágil con énfasis en modelo de ciclos
Agile RUP	dX	Booch, Martin, Newkirk / 1998	Framework/Disciplina	XP dado vuelta con artefactos RUP
Dynamic Solutions Delivery Model	DSDM	Stapleton / 1997	Framework/modelo de ciclo de vida	Creado por 16 expertos en RAD
Evolutionary Project Management	EVO	Gilb / 1976	Framework adaptativo	Primer método ágil existente
eXtreme Programming	XP	Beck / 1999	Disciplina en prácticas de ingeniería	Método ágil radical
Feature-Driven Development	FDD	De Luca & Coad / 1998- Palmer & Felsing / 2002	Metodología	Método ágil de diseño y construcción
Lean Development	LD	Charette / 2001, Mary y Tom Poppendieck	Forma de pensar – modelo logístico	Metodología basada en procesos productivos
Rapid Development	RAD	McConnell / 1996	Survey de técnicas y modelos	Selección de mejores prácticas, no método
Microsoft Solutions Framework	MSF	Microsoft / 1994	Lineamientos, disciplinas, prácticas	Framework de desarrollo de soluciones
Scrum	Scrum	Sutherland / 1994 Schwaber 1995	Proceso – framework de management	Complemento de otros métodos, ágiles o no

Tabla 3. Convergencias y divergencias entre las principales metodologías ágiles.

Fuente: Metodologías Ágiles [13]

Los beneficios que brindan las metodologías ágiles son innumerables, y actualmente tienen gran acogida en proyectos de desarrollo de software, sin importar que estos sean de pequeño o gran alcance.

Las metodologías ágiles sin duda se acoplan a la realidad cambiante de los negocios que existe en la actualidad; una de estas metodologías es Scrum sobre la que se basará el posterior estudio, analizando el importante aporte que esta hace para lograr reducir riesgos en proyectos de desarrollo de software, además se recalcará que adicional a los beneficios ya mencionados anteriormente, tiene otros característicos de esta metodología, los que se indicarán más adelante. Scrum es una de las metodologías más importantes de su tipo, y así la más popular para la gestión de proyectos; proponiendo así buenas prácticas para el desarrollo de software, que utilizadas de manera correcta y oportuna brindan grandes beneficios a los proyectos, estas acciones deben ser realizadas por el apropiado equipo de trabajo, el cual deberá poseer valores y aspectos positivos como: inter-funcionalidad, auto-organización y comunicación constate; los mismos que sugiere e impulsa esta metodología.

## **CAPÍTULO II.**

### **ANÁLISIS DE SCRUM.**

#### **2.1. CARACTERÍSTICAS DE SCRUM**

“En el año 1995 Schwaber y Sutherland, presentaron en paralelo una serie de artículos describiendo Scrum, durante el OOPSLA '95 desarrollado en Austin, siendo ésta la primera aparición pública de la metodología. En años anteriores fueron estos dos empresarios quienes pusieron en práctica en sus empresas los primeros conceptos de la metodología, es así como a principios de los años 1990 Ken Schwaber empleó una aproximación de Scrum en su compañía, Advanced Development Methods, y por aquel tiempo Jeff Sutherland desarrolló una aproximación similar en Easel Corporation y fue el primero en denominarla Scrum. Durante los años siguientes, Schwaber y Sutherland, colaboraron para consolidar los artículos antes mencionados, así como sus experiencias y el conjunto de mejores prácticas de la industria que conforman a lo que ahora se le conoce como Scrum. En 2001, Schwaber y Mike Beedle describieron la metodología en el libro Agile Software Development with Scrum.”(Scrum [26])

El criterio ágil ha tenido mucha acogida en los últimos años, y Scrum es una de las metodologías ágiles más usadas actualmente. Si bien muchos autores no toman a Scrum como una metodología, sino más bien como un conjunto de buenas prácticas, esta cumple con todas las características del manifiesto ágil. Una de las características de Scrum es la flexibilidad, tomando en cuenta que lo más adecuado debería ser que la metodología se adapte a la empresa y no el caso contrario, de esta forma siendo Scrum una de las metodologías de fácil de adaptación a las empresas, además de no aplicar métodos excesivamente estrictos como las metodologías tradicionales, esta metodología también posee la característica de requerir un equipo multidisciplinario, lo que permite enfrentarse a los diferentes retos que se presentan durante el proyecto,

esto y muchos otros aspectos han logrado que esta metodología vaya aumentando su popularidad radicalmente en los últimos años.

### **2.1.1. CARACTERÍSTICAS.**

- **La incertidumbre.**

En los diferentes proyectos de software es muy común que en primera instancia no se obtenga una visión específica y total del producto a ser desarrollado, ya que en el transcurso del proyecto, esta pueda sufrir variaciones, debido a que el entorno que manejan la mayoría de casos es inconstante, por lo que es de ayuda primero obtener una visión general, ampliando el campo de comprensión a medida que se desarrolle el proyecto, esto permite una mayor libertad y autonomía, en la toma de decisiones y en las posibles innovaciones que se quiera proporcionar al proyecto, esto con el tiempo logrará transformarse en los requerimientos específicos, que serán solamente comprendidos si se ha partido de algo general, entendido de manera correcta.

- **Equipos de desarrollo auto-organizados.**

La auto organización no quiere decir lo mismo que la autodirección, ya que en la primera se cuenta con una autonomía, auto superación y auto enriquecimiento; todo esto le permite a cada miembro del grupo poder tomar decisiones oportunas, y brindar soluciones correctas y valiosas que aportan a cada miembro del grupo; en cambio que en la autodirección, las decisiones tomadas solo afectan al miembro que realice dicha acción, ocurriendo que en algunos casos solamente se actúe en beneficio propio del individuo, y no del grupo de trabajo.

- **Fases de desarrollo solapadas.**

Lo que en el desarrollo tradicional es llamado fase, para Scrum es una actividad, pudiendo ser realizadas según las necesidades de cada iteración, en el orden requerido, e incluso compartiendo tiempos de ejecución.

Debido a que cada iteración va tomando su propia forma según las necesidades, el añadir un nuevo requerimiento es de alto valor para el proyecto, permitiendo así la retroalimentación, y ayudando a los miembros del trabajo a esclarecer y puntualizar la visión del producto. En Scrum cada iteración es conocida como **sprint**.

- **Control sutil.**

Si bien el grupo de trabajo en Scrum se caracteriza por la autonomía con la que puede ejecutar sus responsabilidades, pero esta no quiere decir que no exista un control sobre el trabajo, sin embargo, a diferencia que otro tipo de metodologías, este se realiza de forma sutil, es decir, que solo habrá la suficiente intervención para que no se cree un ambiente de tensión, y en donde se puede desarrollar la espontaneidad y creatividad.

Para que el líder del grupo pueda lograr ejercer un control sutil sobre este, es necesario que el pueda confiar en cada miembro, para esto el debe estar seguro de las habilidades de cada persona, por lo que es necesario tomar el tiempo preciso para seleccionar a cada miembro, con lo cual se podrá garantizar que se forme un buen equipo y confiable. Con la ayuda de cada miembro se debe crear un ambiente abierto, para que cada miembro del grupo tome la iniciativa de involucrarse con las necesidades del cliente, en este contexto es importante que la persona que ejercerá el control, tenga la capacidad para predecir y tolerar los errores, los cuales siempre existirán. Este tipo de control para Scrum se lo realiza por medio de breves reuniones en las cuales se revisa el trabajo ya realizado y el que está por realizarse.

- **Difusión y transferencia del conocimiento.**

La difusión de conocimiento debe realizarse en todo sentido, es decir, entre los miembros del grupo, y desde ellos hacia el cliente y viceversa, recuerde que cada uno de ellos pueden aportar información muy valiosa, basándose no

solamente en sus conocimientos sino en experiencias, lo que lleva a que cada persona gane conocimiento único, convirtiendo al grupo en multidisciplinario, y garantizando la transparencia de la información.

## **2.1.2. ELEMENTOS DE SCRUM.**

### **2.1.2.1. Roles.**

En el grupo de trabajo debe estar claro cómo funciona Scrum, y cuáles son las características por las cuales se guiarán el desarrollo del producto, para que de esta manera sea posible realizar fácilmente las responsabilidades asignadas.

Los roles que maneja Scrum son los siguientes:

- **Propietario del producto.**

Esta persona tiene como responsabilidad brindar la mayor información, para lograr el producto final deseado, ya que es el que se encuentra más familiarizado con el negocio, además con él se tratará la financiación del proyecto. Esta información proporcionada dará como resultado el Producto Backlog.

- **Scrum Manager.**

Es el responsable del funcionamiento de Scrum en el proyecto, y de lograr que las prácticas de trabajo de esta metodología se acoplen a la empresa y no viceversa, para que de esta manera se alcance el producto final deseado.

Puede ser muy aconsejable delegar este rol a una persona que sea parte del negocio, la cual debe recibir la formación adecuada para desempeñar sus responsabilidades, acoplándose a la gestión y desarrollo ágil. Esta opción deberá analizarse según los requerimientos del proyecto, ya que en otras ocasiones podría funcionar que este rol lo desempeñe una persona que ya sea parte del proyecto y tiene conocimiento sobre Scrum.

- **El equipo.**

El equipo es el responsable del desarrollo del producto, en el también es tomado en cuenta el propietario del producto, que a diferencia de otras metodologías el cliente es tomado como algo independiente del grupo. En Scrum es de suma importancia que el cliente contribuya al proyecto, cumpliendo sus responsabilidades, y convirtiéndose así, en una parte vital del proyecto.

El equipo tiene carácter multidisciplinario, lo que le permite poseer la capacidad de auto organizarse, y adquirir las atribuciones necesarias y oportunas para la toma de decisiones.

#### **2.1.2.2. Documentos requeridos en Scrum.**

Como resultado de las reuniones se obtiene una lista de requerimientos, los cuales posteriormente se transforman en tareas a ser realizadas por los miembros del grupo. Estos listados no son documentos definitivos o estáticos, ya que irán evolucionando a medida que el proyecto avance, pero sí contienen una visión clara y amplia de lo expuesto en cada uno.

A continuación se detalla cuales son los documentos en los que se presentarán los requerimientos del sistema:

- **Product Backlog.**

Este listado contiene los requisitos del sistema, con la prioridad respectiva, siendo así la visión deseada del producto, la cual evolucionará con el tiempo, convirtiéndose en un histórico de los requerimientos que se implementarán.

Este documento se genera a partir de una reunión que se realizará para tener una visión clara y total del sistema. Aportarán todos los miembros del grupo de trabajo, compartiendo sus puntos de vista sobre el producto a ser desarrollado, los mismos que deberán partir de la visión real del producto, debiendo quedar entendida claramente por cada miembro del grupo de trabajo.

El responsable de este listado, la metodología Scrum lo denomina "Propietario del Producto", o en otras palabras cliente, que es el que tiene mayor conocimiento y relación sobre las necesidades del negocio que se desean cubrir con el sistema. "El Product Backlog se sitúa en el área de requisitos o necesidades de negocio desde el punto de vista del cliente."(Flexibilidad con Scrum [5])

- **Sprint Backlog.**

Este documento contiene los requerimientos que el equipo se compromete a realizar en el sprint próximo a ejecutarse, entendiendo así que existirá un documento de este tipo por cada iteración que se desarrolle en el proyecto. El listado además de los requerimientos, que deberán estar priorizados, contendrá las tareas necesarias para cubrir la funcionalidad del sprint, y de esta manera cumplir con el objetivo de la iteración, con esta información se asignarán responsables y tiempos de ejecución. Cada tarea realizada pasará a formar en la iteración, una parte completa y funcional del producto.

"El Sprint Backlog se sitúa en el área de especificación de los requisitos de software necesarios para dar respuesta a las funcionalidades esperadas por el cliente." (Flexibilidad con Scrum [5])

Dado que cada Sprint Backlog solamente contendrá aquellas tareas que se ejecutarán en cada iteración, es de suma ayuda que para la planificación y control, el equipo sea consciente de lo que puede y posteriormente deba realizar en el sprint a ser desarrollado, con lo que podrá gestionar de mejor manera los posibles riesgos.

En Scrum no se maneja el control riguroso, ya que no es necesario recordar constantemente a los miembros del equipo cuáles son sus tareas y responsabilidades, para esto se puede ayudar visualizando el Sprint Backlog físicamente las tareas en una pizarra o pared en el lugar de trabajo, de manera

que para cada persona sea de fácil acceso, y así constituir en un recordatorio personal a cada miembro del grupo.

- **Incremento.**

Este es el resultado de cada sprint, en el cual se presenta una parte del sistema que ya es funcional para el usuario, y que se encuentra terminada con las pruebas y la documentación requerida.

### **2.1.2.3. Reuniones.**

Es importante recalcar que para cada sprint o iteración se realizará la planificación, seguimiento y revisión; que ayudarán a llevar el control necesario y la retroalimentación respectiva del proyecto. A continuación se detallan cada uno de los procesos antes mencionados:

- **Planificación.**

Para cada sprint se debe definir el objetivo y el trabajo que se realizará para cumplirlo, de esta manera cubrir los requerimientos establecidos en cada iteración, esta planificación además toma en cuenta las prioridades del negocio, que se acordarán con el cliente.

La planificación se realizará en una reunión formal, a la que deben asistir el propietario del producto, los miembros del grupo de trabajo, y de ser necesario otras personas que pueden estar involucradas en el proyecto. Para realizar esta reunión se debe tener previamente desarrollado el Product Backlog, que es el listado de requerimientos totales que se cubrirán en el proyecto, partiendo de este documento, se realiza una revisión sobre este listado, para saber si existe alguna variación significativa sobre este, y poder registrarla en dicho documento. La reunión puede tomar como máximo un día, el responsable de organizarla es el Scrum Manager, pueden asistir todas aquellas personas que crean conveniente su participación, para que brinden la información necesaria, se

realizará en dos partes, con la pausa respectiva; en la primera se determina que requerimientos del Product Backlog se desarrollarán en la iteración, lo que comprenderá el objetivo o meta del sprint; esto puede tomar como máximo cuatro horas; en la segunda parte enfatizando en los requerimientos que contendrá el sprint a ser ejecutado, se define las tareas que se deben realizar para cumplir el objetivo del sprint. Ya con el conocimiento de lo que se irá a desarrollar en dicha iteración, se podrán estimar esfuerzos, responsables y tiempos de ejecución.

Para llevar a cabo esta reunión es importante que el grupo de trabajo tenga una idea clara de las necesidades que desea cubrir el cliente con el proyecto, y cuales son de mayor prioridad, también deberán saber sobre la tecnología a ser usada, de esta manera teniendo conocimiento tanto del negocio como de las herramientas que se utilizarán, se puedan generar juicios validos para el proyecto, es decir, juicios de expertos. El producto de esta reunión es llamado Sprint Backlog, que contendrá el objetivo del sprint, la lista de tareas con sus prioridades respectivas, y la duración total de la iteración.

A continuación se cita como procederá la reunión:

- Por parte del propietario del producto se presenta el Product Backlog, y enfatiza cuales son los requerimientos que a su criterio deben priorizarse en el sprint a ejecutarse. Esta información debe ser expuesta con el suficiente detalle, para que quede totalmente entendida, y clara para el público asistente.
- El grupo de trabajo se procederán con las preguntas y dudas respectivas, así como el planteamiento de soluciones de posibles diferencias o problemas existentes. Esta etapa es de vital importancia ya que es donde de intercambiará toda la información necesaria para lograr un consenso, y de esta manera acordar el objetivo del sprint. Luego de esto es muy aconsejable que se realice una pausa.

- En la segunda parte de la reunión, el equipo de trabajo define las tareas que involucran el Sprint Backlog, con sus respectivos tiempos y responsables, para esta etapa el propietario del producto solo participa de forma que pueda asegurarse que el equipo ha entendido cual deberá ser el objetivo del sprint, el Scrum Manager desempeñará el papel de moderador para esta etapa de la reunión.

- **Seguimiento.**

El seguimiento de la iteración se la realizará diariamente, en una reunión que durará 15 minutos, para facilitar la visión del avance de cada sprint, en la cual se responde a tres preguntas:

- ¿Cuál fue el trabajo realizado desde la reunión anterior?
- ¿Cuál será el trabajo que se realizará hasta la próxima reunión?
- ¿Cuáles son los obstáculos o problemas que se han presentado creen que se podrán presentar? Para los cuales se debe buscar la solución más óptima

Cuando esta información se tiene clara se puede ir actualizando el Sprint Backlog o listado de requerimientos del sprint, con las tareas ya terminadas.

A esta reunión asisten el Scrum Manager y el resto de los miembros del equipo, de los cuales uno debe tomar el papel de Team Leader, quien garantiza que los tiempos y formatos sean cumplidos. La información que se usa para el desarrollo de la reunión son: el Sprint Backlog y gráfico burn down, que es el que indica el avance del sprint según lo planificado, estos documentos deben estar actualizados desde la última reunión.

- **Revisión.**

En esta reunión se presenta el resultado del sprint al cliente y otras personas involucradas, la misma que debe durar como máximo 4 horas. Esta reunión es de

vital importancia para la retroalimentación del proyecto, y a la vez ayuda a definir y aclarar ideas, tanto en la parte que se refiere al propietario del producto como al Scrum Manager, lo que ayudará para la ejecución de la próxima iteración.

Como resultado del sprint deben presentarse todas las funcionalidades acordadas cuando se inicio la iteración, siendo esta una reunión informativa la cual no servirá para realizar críticas al sistema.

En la exposición solo debe presentarse el sistema sin otro tipo de ayuda o presentaciones digitales, y no debe tomar más de una hora en la demostración de la funcionalidad, según se vea conveniente se puede incluir documentación.

Se puede seguir el siguiente orden para la presentación:

- El Scrum Manager realiza una presentación breve sobre las funcionalidades del Product Backlog para esta iteración.
- Se presenta el sistema con la parte operativa acordada hacia el cliente.
- Una vez presentado el sistema se puede tomar un tiempo para sugerencias y preguntas.
- Se acuerda con el propietario del producto cuando se realizará la presentación del siguiente sprint.

## **2.2. HERRAMIENTAS UTILIZADAS EN SCRUM.**

### **2.2.1. GRÁFICO BURN-UP.**

Es una herramienta para el propietario del producto, que permite realizar el seguimiento respectivo durante el proyecto. Para esto se realiza un gráfico en donde el eje y estará representado el esfuerzo del grupo de trabajo, y en el eje x se representará las fechas estimadas de entrega para cada sprint.

El esfuerzo del eje y se puede expresar en punto de función o en unidades relativas (tiempo teórico), además se marcarán las versiones que se van a presentar.

A continuación se cita un ejemplo del gráfico Burn-Up, en la Figura 1:

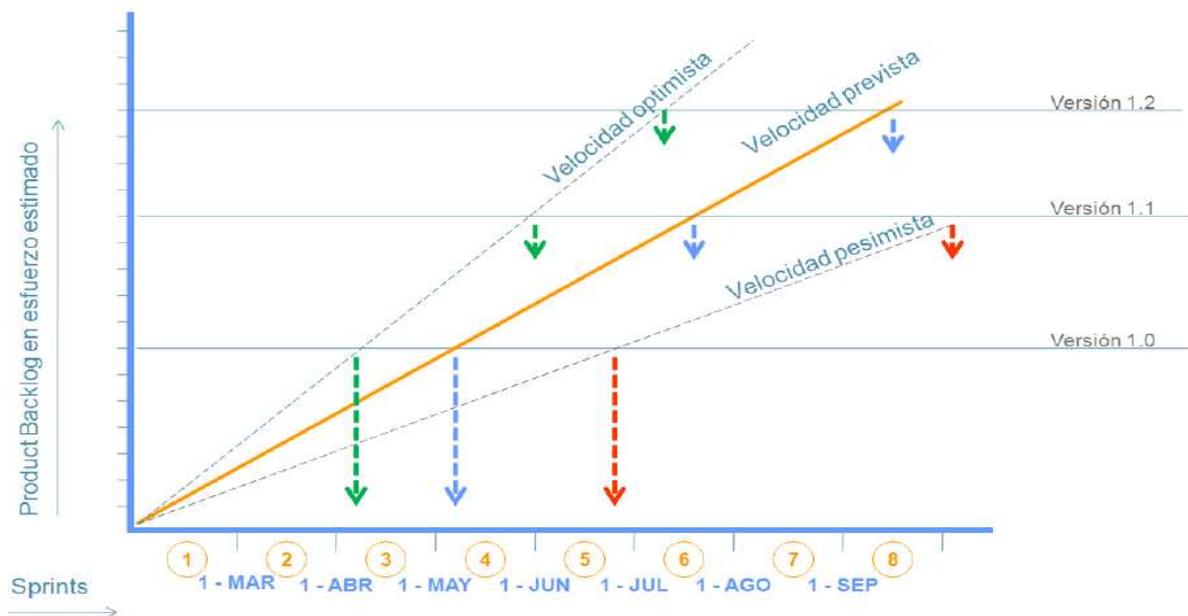


Figura 1. Gráfico Burn-Up.

Fuente: Flexibilidad con Scrum [5]

Según los resultados que se visualicen en la gráfica, se podrá revisar y verificar si el ritmo de avance del proyecto se encuentra según lo previsto. Debido a que esta gráfica abarca todas las iteraciones del proyecto, se puede detectar fácilmente desde los primeros sprints si existe algún inconveniente que retrase el proyecto, y de esta manera poder realizar las acciones correctivas y lograr la velocidad prevista.

### 2.2.2. GRÁFICO BURN-DOWN.

Es una herramienta que servirá al equipo de trabajo para realizar el respectivo seguimiento a cada sprint, y detectar posibles inconvenientes.

En el gráfico se representa en el eje y el esfuerzo que toma en realizar determinadas tareas, y en el eje x el tiempo que durará el sprint. Se espera que no existe desviaciones en esta grafica, pero de existir será un indicador de que las estimaciones iniciales no fueron las correctas.

En el Figura 2 de ejemplo, la línea recta representa la velocidad estimada para el sprint, y la otra simboliza las desviaciones de velocidad que existieron durante el desarrollo del sprint.

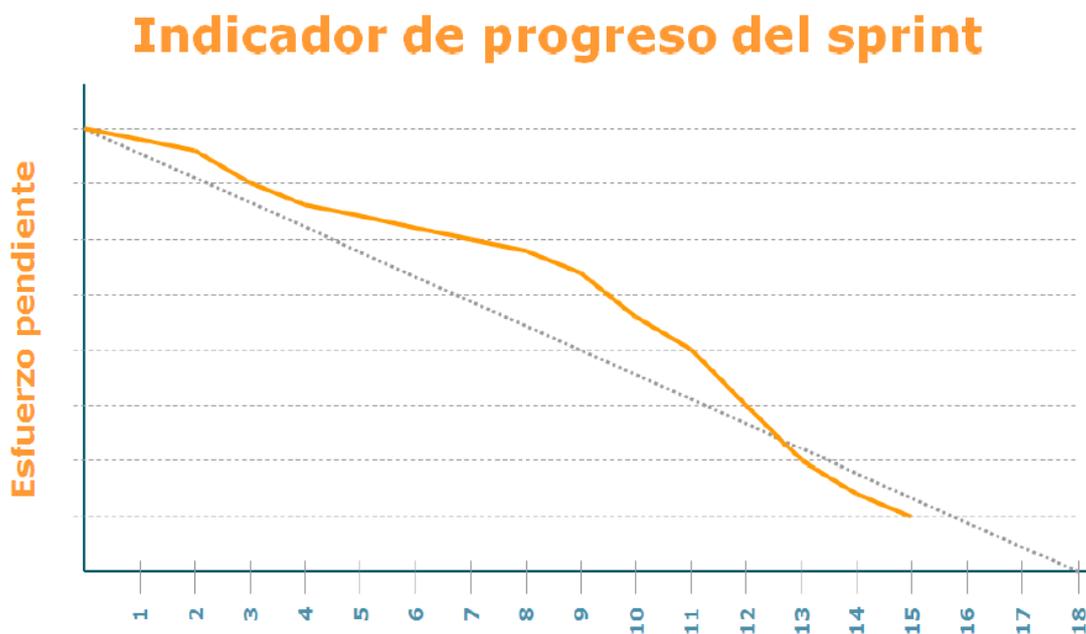


Figura 2. Gráfica Burn-Down.

Fuente: Flexibilidad con Scrum [5]

### 2.2.3. JUEGO O PROTOCOLO DE DECISIÓN - ESTIMACIÓN DE PÓKER.

En este protocolo de toma de decisiones, se reduce significativamente el tiempo que se toma para determinar la duración de las tareas, ya que se realiza un acuerdo con el conocimiento experto de los participantes del grupo de trabajo.

A continuación se detalla cómo se debe proceder:

- Se debe analizar y entender perfectamente cuales son las tareas a ser estimadas su esfuerzo o tiempo de duración.
- Se estima una escala mínima y máxima para la duración de cada tarea, se recomienda que la unidad mínima sea tomada en base a la duración total del sprint, que podría ser medio día.
- De acuerdo a la escala, se seleccionarán cual será la cantidad de días que representará cada carta, tomando 8 cartas como máximo ya que la de mayor valor deberá contener el símbolo infinito, esta representará el número de días máximo seleccionados anteriormente.
- Cada participante con sus cartas indicará por cada tarea el número de días que según su criterio cree que deberá durar la tarea, luego se tomará como valor final la carta que tenga mayor repetición, de no existir se promediará el número de días seleccionado por cada integrante, esto se realiza siempre y cuando exista el consentimiento de los participantes. En el caso de existir una sobre estimación o sub estimación del tiempo, el team leader se encargará de cuestionar la razón para haber seleccionado dicho valor, de no llegar algún acuerdo en alguna tarea, esta se deberá analizarla posteriormente.

Cuando existan tareas donde se piensan que durarán varios días más allá del valor tomado como máximo, se podrá pedir al propietario del producto que revise nuevamente la tarea y busque la manera de descomponerla o reducir su alcance.

En la Figura 3, se detalla un ejemplo de las cartas que se pueden utilizar para el protocolo de decisión.

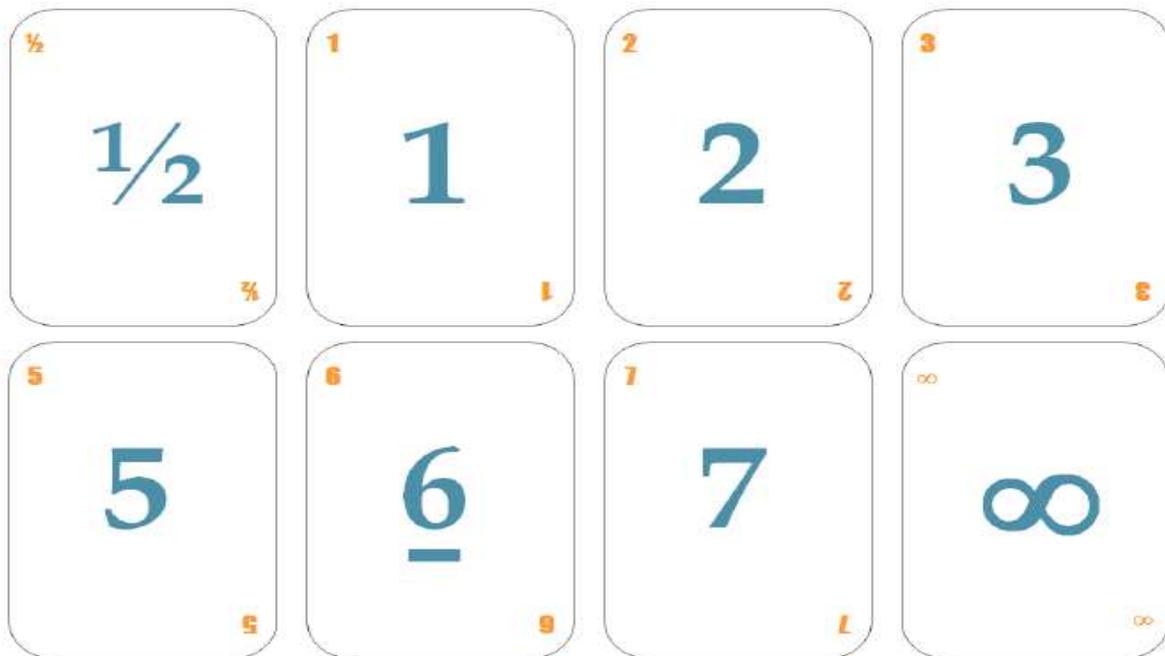


Figura 3. Juego o Protocolo de decisión – Estimación de Póker.

Fuente: Flexibilidad con Scrum [5]

#### 2.2.4. DESARROLLO DE RETROSPECTIVA.

Esto es usado básicamente para analizar los problemas que se han presentado durante un sprint y poder realizar acciones correctivas sobre estas, tomando en cuenta que estas no deben tomar un esfuerzo excesivo, ya que esto podría demorar el desarrollo del proyecto.

Los pasos a seguir para realizar la retrospectiva se detallan a continuación:

- En un pizarrón se trazara una línea horizontal que equivale al tiempo que duró el sprint, esto ayuda a ubicar los eventos en el tiempo que sucedieron, por lo que se deberá tomar una escala que represente ciertos periodos de tiempo en el sprint.

- Cada miembro del equipo colocará en la parte superior los aspectos o sucesos positivos, y en la parte inferior los negativos. Se recomienda que como máximo se citen 3 eventos por cada persona del grupo.
- Realizado el análisis correspondiente se tomarán los 3 problemas más importantes, el resto se puede dejar para revisarlos posteriormente y verificar si continúan sucediendo en las siguientes iteraciones, esto representará al primer listado de ideas.
- De los tres problemas seleccionados anteriormente, se debe analizar las causas de cada uno y buscar las razones del por qué se produjeron, tomando 3 ideas como máximo por cada uno, esto constituirá el segundo listado de ideas.
- Para el tercer listado de ideas se deben buscar las acciones que brindarán la o las soluciones más óptimas para cada uno de los problemas, tomando en cuenta que estas no deben tomar un esfuerzo exagerado ni tampoco mínimo, ya que esto podría ocasionar que la solución planteada remedie simplemente el problema temporalmente, y no de forma definitiva.

#### **2.2.5. DESARROLLO DE DOCUMENTO PRODUCT BACKLOG.**

Esto es considerado una herramienta de seguimiento para el equipo. En este listado se pueden incluir los parámetros que el grupo crea conveniente ya que este no es un documento completamente estricto ni cerrado.

Los siguientes son parámetros que se pueden aplicar al formato:

- **Indicador único de la funcionalidad:** esto puede ser de mucha ayuda al momento de referirnos a la funcionalidad, ya que solo se podría indicar el número y no el nombre completo, que en algunos casos podría ser muy largo.
- **Descripción de la funcionalidad:** se debe indicar la información que sea necesaria, evitando alargar con datos que no son de mucha ayuda.

- **Nivel de priorización:** esto lo debe indicar el Propietario del Producto, ya que él es quien está más familiarizado con las necesidades del negocio.
- **Estimación de esfuerzo:** esta se podrá indicar en base a la medida estándar que haya seleccionado el grupo, podría ser tiempo o puntos de función.
- **Número de sprint en el que se realizará:** esto indicará en sprint en el que se desarrollará la funcionalidad.
- **Responsable de la funcionalidad:** será importante que el responsable se indique en este documento, ya que puede evitar posibles confusiones futuras; en el caso de existir más de uno, se deben indicar todos los responsables asignados a dicha funcionalidad.
- **Observaciones:** se deberá colocar la información adicional que se crea que pueda ayudar, evitando información que no sea lo suficientemente clara y de poco.

A continuación se indica en la Tabla 4, como quedaría el formato de Product Backlog:

ID	DESCRIPCIÓN	PRIORIZACIÓN	ESTIMACIÓN ESFUERZO	RESPONSABLE	SPRINT	OBSERVACIÓN
1						
2						
3						
4						
5						

Tabla 4. Formato de Product Backlog.

Autor: Lilián Elizabeth Arroba Medina.

### 2.2.6. DESARROLLO DE DOCUMENTO SPRINT BACKLOG.

Es una herramienta colaborativa y de gestión para el grupo de trabajo durante la realización de un sprint. Los parámetros del formato se pueden modificar según las necesidades del proyecto, puede cubrir la siguiente información:

- **Identificador de la funcionalidad a la que pertenece:** esta información se obtiene del documento del Product Backlog, con esto se logra identificar el requerimiento de la misma forma durante todo el proyecto.
- **Nombre explicativo de la tarea:** este deberá solo contemplar la información necesaria, para evitar usar nombres demasiado extensos, que puedan crear confusión.
- **Responsable:** de deberá indicar el o los responsables que cubrirán la tarea, para que tengan presente sus compromisos. Este campo debe tener correspondencia con los responsables asignados en el Product Backlog.
- **Duración:** realizados los acuerdos y análisis necesarios, se debe indicar el tiempo que tomará cubrir cada tarea.
- **Estado:** para poder llevar un control y seguimiento, se pueden utilizar estados que indiquen el progreso de las tareas, y comprobar si se desarrollan según lo planteado y acordado.

Este listado debe tener solamente la información estrictamente necesaria, la forma que se seleccionará para ser visualizada dependerá solamente del grupo, y como este pueda facilitar el trabajo y la comunicación, por lo que se puede usar una hoja de cálculo o algún medio físico donde se pueda visualizar este listado.

A continuación se detallan algunos ejemplos de cómo se puede representar el Sprint Backlog, el cual siempre se podrá ajustar a las necesidades del proyecto.

<b>SPRINT: 1</b> <b>FECHA DE INICIO: 01 Octubre</b> <b>2010</b>				<b>1-Oct</b>
				<b>Tareas pendientes:</b> 20
				<b>Horas pendientes:</b> 300
FUNCIONALIDAD PRODUCT BACKLOG	DESCRIPCIÓN DE LA TAREA	RESPONSABLE	ESTADO	DURACIÓN
1	Descripción de la tarea 1		Terminada	18
1	Descripción de la tarea 2		Terminada	20
1	Descripción de la tarea 3		Pendiente	10
2	Descripción de la tarea 4		Pendiente	3
2	Descripción de la tarea 5		Pendiente	6

Tabla 5. Representación de Formato de Product Backlog – Hola electrónica.

Autor: Lilián Elizabeth Arroba Medina.

<i>Producto</i>	<i>Pendiente</i>	<i>En curso</i>	<i>Terminada</i>
Descripción Funcionalidad	Tarea Responsable Duración		
Descripción Funcionalidad	Tarea Responsable Duración	Tarea Responsable Duración	
Descripción Funcionalidad	Tarea Responsable Duración		Tarea Responsable Duración

Figura 4. Representación de Formato de Product Backlog – Pizarrón.

Autor: Lilián Elizabeth Arroba Medina

## 2.3. DESCRIPCIÓN DEL FUNCIONAMIENTO Y BENEFICIOS DE SCRUM.

### 2.3.1. FUNCIONAMIENTO DE SCRUM.

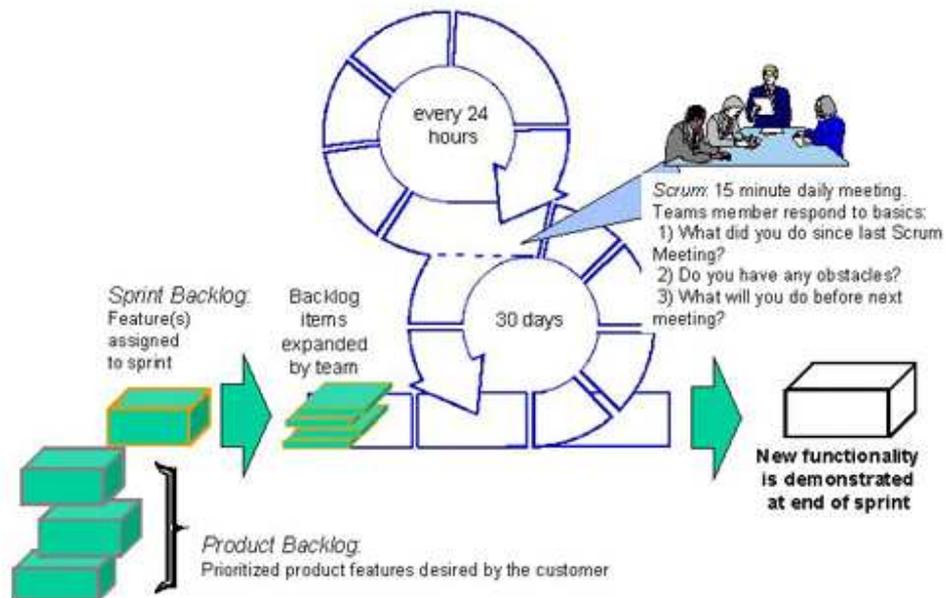


Figura 5. Metodología Scrum – Proceso.

Fuente: Metodologías Ágiles (introducción) [14]

El éxito de usar Scrum en proyectos de desarrollo de software para cualquier organización, depende del compromiso que esta tenga para poder trabajar con dicha metodología, y hacer de sus prácticas parte de la organización, proveyendo así los recursos necesarios para brindar a los miembros del negocio los conocimientos adecuados para que se familiaricen con las prácticas ágiles, todo esto se debe realizar sin tener que modificar drásticamente las políticas propias del negocio.

El rol del Scrum Manager es de vital importancia para el proyecto, ya que una de las responsabilidades es lograr que exista armonía entre Scrum y la organización, el cual debe realizar monitoreo periódico para garantizar la mejora continua sin olvidar

incentivar a la auto organización en todos los miembros del grupo incluido el cliente, tomando sus tareas no simplemente como un rol, sino como responsabilidades a cumplir, las cuales se facilitan enormemente si se asignan estas a las personas más adecuadas.

En Scrum se manejan iteraciones cortas y fijas, con una duración que puede variar desde 1 mes como máximo y dos semanas como mínimo, en cada una se deberá entregar un producto ya funcional para el cliente, con lo que el negocio ya pueda ver los beneficios del proyecto en las primeras iteraciones.

Para iniciar el proyecto se debe previamente realizar una reunión entre las personas involucradas, para poder tener una idea acerca del alcance que tendrá el proyecto.

El funcionamiento de Scrum se describe a continuación:

- **Primera etapa: Definición del Product Backlog.**

Para esto, se realizará una reunión en la etapa inicial del proyecto, a la cual deben asistir todos los miembros del grupo y los miembros interesados del negocio, en la cual se definirá la lista de requerimientos, que en Scrum se define como Product Backlog, en esta se deberán priorizar los requerimientos, de forma que aporte con beneficios para la empresa en las primeras etapas del proyectos, y contribuya al éxito del proyecto. La información que contendrá este documento debe ser provista por las personas involucradas en el negocio de forma clara.

Es importante recalcar que esta lista en la mayoría de los proyectos desarrollados con Scrum, se convierte en dinámica, es decir esta podrá ir variando durante el tiempo de vida del proyecto, convirtiéndose así en un histórico de los posibles cambios.

- **Segunda etapa: Definición del Sprint Backlog.**

Como segunda parte, luego de la reunión donde se definió la lista total de requerimientos del proyecto, es decir el Product Backlog, se realizará una reunión donde se especificará el objetivo del sprint y la respectiva planificación. Para cada iteración se realizará una reunión independiente, en la cual se planificará únicamente la meta del sprint, este proceso de organización se debe realizar previo a la iniciación de cada iteración.

Para poder definir el objetivo del sprint, se partirá del Product Backlog, de esta lista se seleccionará de acuerdo a la prioridad y tamaño de los requerimientos, cual o cuales se cubrirán en el sprint próximo a ejecutarse, para lo cual se debe tener un previo conocimiento del negocio para poder llegar a un acuerdo con el cliente, de esta forma aportar con beneficio tanto para la empresa como al proyecto.

- **Tercera etapa: Definición de Tareas y Responsabilidades.**

Una vez definido el Sprint Backlog con los requerimientos seleccionados de acuerdo a las prioridades del negocio, como parte de la planificación del sprint, se deben plantear las tareas a ser realizadas para cumplir cada requerimiento y de esta manera finalizar el objetivo del sprint, con la lista de tareas planteada, se asignarán los responsables para cada una, y los tiempo de ejecución de las mismas. Para esto se puede usar alguna herramienta de decisión como la estimación de Poker, mencionada anteriormente, para esta parte de la planificación, es de vital importancia que se realice en base a la experiencia y conocimiento de los miembros del grupo.

- **Cuarta etapa: Seguimiento de la iteración.**

Para el seguimiento o monitoreo de la iteración podemos usar algunas herramientas propias de Scrum que facilitan el control en cada sprint:

- Reunión Diaria.

Esta reunión se debe realizar todos los días, con una duración de 15 minutos, en la cual se debe realizar las siguientes interrogantes a cada miembro del grupo:

- ¿Cuál fue el trabajo realizado desde la reunión anterior?
- ¿Cuál será el trabajo que se realizará hasta la próxima reunión?
- ¿Cuáles son los obstáculos o problemas que se han presentado creen que se podrán presentar?. Para los cuales se debe buscar la solución más óptima.

- Grafico Burn-Down.

Este gráfico explicado anteriormente en la sección 2.2, es el que permite visualizar el progreso del sprint, y poder detectar posibles inconvenientes.

- Sprint Backlog

El listado de requerimiento o Sprint Backlog, para poder manejarlo de mejor forma, es recomendable visualizarlo de forma que todos los miembros del grupo tengan presente sus tareas y responsabilidades, y además se pueda actualizar fácilmente, es por eso que el grupo de trabajo deberá decidir si se representa en una hoja electrónica o en un pizarrón.

- **Quinta etapa: Presentación de la iteración.**

Para esto se realizará una reunión en donde todos los miembros del proyecto y las personas interesadas del negocio deben asistir para poder aportar con las opiniones respectivas, y de esta manera contribuir para la retroalimentación del proyecto.

En esta reunión se realizará la presentación del resultado de cada iteración, en la cual el cliente podrá constatar que lo acordado inicialmente como objetivo del sprint, se ha cubierto de manera óptima, además el propietario del producto deberá realizar el Gráfico Burn Up, que se explico en la sección 2,2, el cual se actualizará a medida que el proyecto avance.

Como segunda parte de esta reunión, se debe realizar una retrospectiva al trabajo realizado por los miembros del grupo, y de esta manera recalcar y analizar las fortalezas y debilidades del grupo de trabajo y del proyecto, para poder ir mejorando cada aspecto que pueda perjudicar al éxito del proyecto.

- **Sexta etapa: Continuidad del ciclo Scrum.**

Esta última etapa solamente indica que desde la segunda etapa hasta la sexta debe ejecutarse para cada sprint o iteración del proyecto.

### **2.3.2. BENEFICIOS DE SCRUM.**

Scrum brinda muchos beneficios a los proyectos de desarrollo que utilizan esta metodología, entre estos se menciona los siguientes:

- Debido a la adaptabilidad que maneja Scrum, los cambios que se generan en los proyectos, se convierten en innovaciones, esto agrega mucho valor al producto final, cubriendo las necesidades del cliente y del mercado actual que es de alta competitividad.
- Al brindar resultados de forma anticipada, sin requerir que el proyecto llegue a su etapa final, provee al cliente un producto que ya puede ser usado, y de esta manera pueda ir recuperando su inversión de forma temprana, lo que a la vez genera mayor motivación en el grupo, ya que esto crea un sentimiento de satisfacción, al entregar resultados de forma temprana en el proyecto.
- Scrum promueve la comunicación continua, lo que ayuda a conocer el desarrollo del proyecto en todo momento, y de esta manera brindar la mejor solución, en el caso de existir inconvenientes; además de poder compartir información necesaria que ayude al proyecto.
- El grupo de trabajo por tener la característica de ser multidisciplinario, con habilidades y conocimientos diversos, ayuda a tomar las decisiones más

oportunas y correctas; por ejemplo en la planificación, cuando se requiere estimar esfuerzo para las tareas, o cuando se requiere plantear soluciones.

- Scrum propone buenas prácticas a los miembros del grupo del proyecto, como: auto-gestión, innovación, adaptabilidad; lo que ayuda a que el proyecto se alinee con las metas del negocio, y así se cree en el cliente la satisfacción de un trabajo bien hecho.
- La retroalimentación que se maneja en Scrum, genera mejora continua en proyectos que son desarrollados con esta metodología.
- Debido a que las iteraciones en Scrum se manejan de forma fija, el compromiso entre los miembros del grupo se logra de manera más concreta, ya que ellos propondrán solo lo que pueden cumplir para el lapso de tiempo acordado, y al manejar iteraciones que no sobrepasaran un mes, ayuda a que el entorno de cada sprint sea más manejable, para la identificación temprana de problemas y como resultado la máxima reducción de riesgos en el proyecto.
- La inclusión del cliente, como parte del grupo de trabajo, ayuda a que este sienta mayor compromiso con el proyecto, y de esta manera se logra la generación del trabajo conjunto, para alcanzar un producto de alta calidad, esto además promueve la continua comunicación con el cliente, ayudando a gestionar las necesidades del negocio, para poder satisfacer sus requerimientos de la forma más adecuada.

### **2.3.3. LIMITACIONES DE SCRUM.**

Entre algunas limitaciones que puede presentar Scrum en proyectos de desarrollo de software, se citan las siguientes:

- Scrum no puede gestionar aquellos riesgos que son considerados externos al proyecto, como aspectos: políticos, legales, o de mercado; que ya no depende de la empresa, para mantener un control.

- Para algunos desarrolladores, Scrum no brinda la guía suficiente para los proyectos de software, debido a su gran apertura para agregar los criterios que sean necesarios al proyecto, sin ninguna limitación.

## **CAPÍTULO III.**

### **DESARROLLO DE PROPUESTA PARA LOGRAR REDUCCIÓN DE RIESGOS EN PROYECTOS DE SOFTWARE UTILIZANDO SCRUM.**

#### **3.1. RIESGOS COMUNES EN PROYECTOS ACTUALES DE PROYECTOS DE SOFTWARE.**

Sería muy difícil emprender un proyecto de desarrollo de software, con la creencia de que este no estará expuesto a riesgos, ya que no existe un ambiente lo suficientemente estable que garantice esto. El mundo actual sufre alteraciones, como: la tecnología y las necesidades comerciales del mercado, que cambian a un ritmo muy acelerado, por lo que si los proyectos de software no van a la misma o al menos similar velocidad, corren el riesgo de sufrir grandes inconvenientes; por eso es importante lograr un equilibrio entre la tecnología, el mercado y todos los elementos que forman el proyecto.

Los inconvenientes en proyectos de desarrollo de software se convierten en un riesgo, y por ende en una amenaza, cuando estos no puedan ser controlados de manera oportuna, y provocan conflictos para lograr el cumplimiento de metas, las mismas que deben garantizar el éxito del proyecto.

Los riesgos en un proyecto de desarrollo de software por lo general se asocian a toda vulnerabilidad, modificando el ambiente de trabajo en el proyecto. Para poder analizar y cuantificar los posibles efectos, es muy importante saber tanto el grado de probabilidad con el que puede presentarse estos conflictos, y los daños, o pérdidas que podrían ocasionar.

Una de las clasificaciones que se pueden hacer a los riesgos, es la que diferencia a estos en: genéricos y específicos; los primeros son aquellos que se presentan de forma

común para todos los proyectos, y los segundos dependen exclusivamente del entorno y ambiente del proyecto.

Los riesgos que se puedan presentar en los proyectos de software podrán originarse en cualquier elemento que forma parte del desarrollo, tornando visibles los posibles problemas en estos elementos, como por ejemplo: el cálculo de presupuesto, planificación, recurso humano, requerimientos, calidad del producto final; comprendiendo estos como riesgos del proyecto, producto, o negocio. Estos también pueden generarse tanto en la parte que corresponde al cliente como en el grupo de desarrollo, incluso podrían existir inconvenientes externos al negocio y al proyecto, para los cuales se deberá contar con el apoyo de estas dos partes, y así brindar la solución más adecuada y oportuna.

Dentro de los posibles problemas que puedan generarse en el negocio podrían citarse: “riesgos de mercado, riesgos estratégico, riesgos de comercialización o venta, riesgos de dirección, riesgos de presupuesto. “(Riesgos del Software [22]) Estos problemas en muchos casos se presentan por falta de una buena y correcta dirección, por personas claves dentro del negocio, las mismas que deben lograr varios aspectos como: motivación, coordinación de actividades, resolución de conflictos, facilitar comunicación, y sobre todo proyectar liderazgo.

En los posibles riesgos que se entienden dentro del proyecto y producto podrían citarse:

- El grupo de trabajo no prevé el avance de la tecnología, queriendo usar alguna que posiblemente en un futuro cercano se encuentre demasiado obsoleta
- Posibles retrasos en la planificación.
- Extremada apertura para que el cliente exponga sus necesidades, aportando información poco necesaria para el proyecto.
- Poca apertura del grupo de trabajo ha alguna tecnología, como herramienta o lenguaje de programación.

- Fechas entrega no realistas para los requerimientos, los que se intentan abarcar en el periodo acordado, que no necesariamente es el adecuado.
- Cambio continuo de los miembros del grupo de trabajo.
- Falta de comunicación con el cliente.
- Gastar esfuerzo del grupo de trabajo en aspectos de mantenimiento, o aspectos que no se planeaban abarcar.

Los inconvenientes pueden ser minimizados si se cuenta con la colaboración de un líder, que aporte valor al grupo y facilite la toma de decisiones, ya que de eso dependerá que se generen diferentes riesgos, entre los cuales se pueden mencionar: incorrecta asignación de responsabilidades, errónea medición de tiempos de ejecución en tareas, equivocada asignación de recursos, equivocada selección de tecnología y herramientas para el desarrollo, selección de personal poco acertada, visión equivocada del alcance del proyecto.

La identificación de los riesgos es de gran importancia, pero esta debe ir acompañada de la búsqueda de la solución la cual debe abarcar: la mitigación, la selección del mejor camino para que dicho riesgo no se vuelva a presentar en el futuro, y la elaboración de un plan de contingencia, por eso no hay que olvidar que una vez identificado el riesgo se debe dar la debida atención para que este no se convierta en un problema con efectos que podrían perjudicar el éxito del proyecto, y por ende la calidad del producto, estas actividades comprenderán la gestión de riesgos, dentro de lo que se deberá dar la priorización respectiva, para la atención y solución correspondiente y de esta manera minimizar o eliminar el impacto.

Uno de los ciclos aconsejables para poder realizar la correcta gestión de riesgos, puede ser el citado en la Figura 6:



Figura 6. Ciclo de gestión de riesgos.

Fuente: Identificación de Riesgos de Proyectos de Software en base a taxonomías [10]

En esta gráfica, el ciclo corresponde al siguiente orden: identificación, análisis, planificación, seguimiento, control; todas estas etapas respetan un orden secuencial, que dependiendo del alcance del riesgo, el tiempo empleado en este ciclo puede variar; también es aconsejable según las necesidades del grupo de trabajo y el nivel de gestión que se aplique a los riesgos, realizar la documentación respectiva, la cual puede servir como un histórico, y respaldo, en el caso de que el riesgo en gestión se presente nuevamente en el futuro del proyecto, o esto puede servir como experiencia a los miembros del grupo, al momento de enfrentarse con inconvenientes similares en otros proyectos de desarrollo; algo que no debe faltar durante la gestión de riesgos, es la comunicación constante, esto siempre ayudará a encontrar la solución más acertada.

En la Figura 7, según Williams Ray, plantea la Gestión de Riesgos, como el cumplimiento de objetivos, basándose en la preparación para la gestión, posteriormente la identificación de riesgos, y finaliza con la mitigación de los mismos.

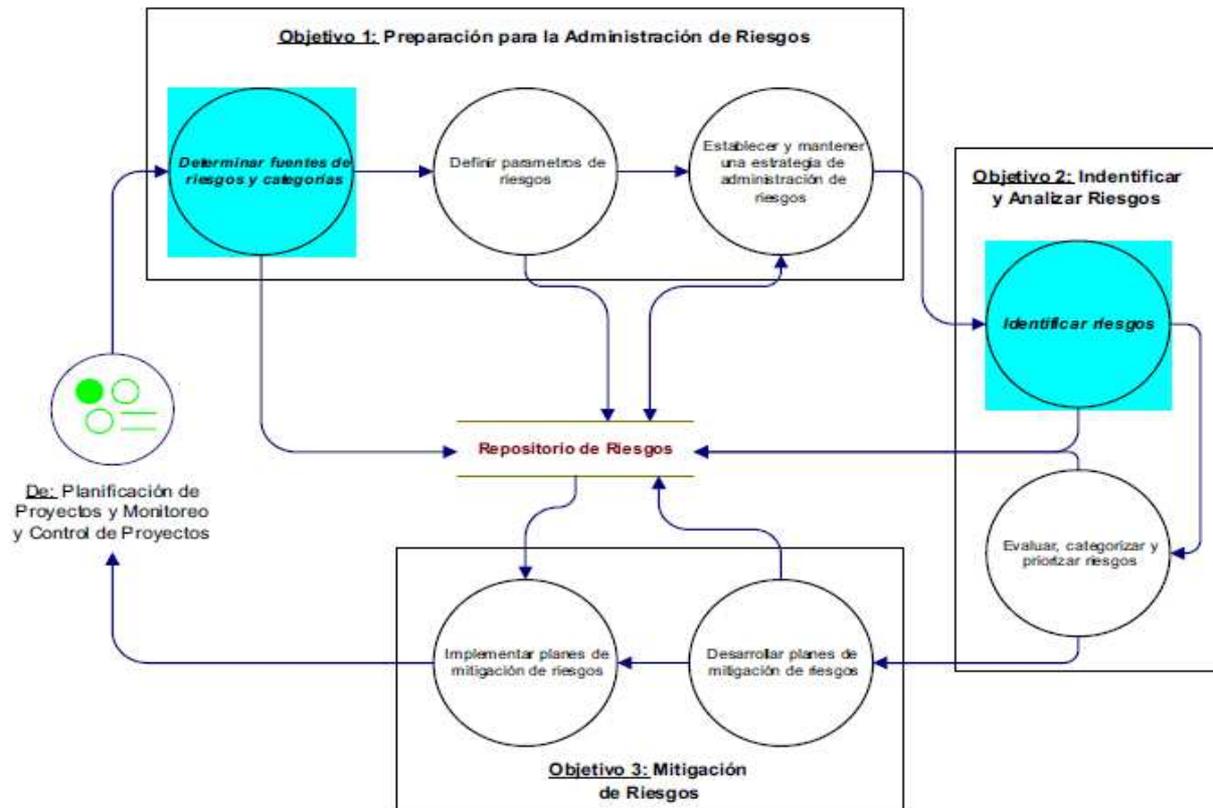


Figura 7. Proceso de gestión de riesgos, basado en la realización de objetivos.

Fuente: Identificación de Riesgos de Proyectos de Software en base a taxonomías [10]

Los riesgos en un proyecto de desarrollo de software no simplemente pueden aparecer en el transcurso del mismo sino previo a la iniciación del mismo que puede ser ocasionado por diferencias de opiniones; también se pueden presentar riesgos en la etapa final durante la fase implantación, lo que puede traer graves inconvenientes y afectar la calidad del producto, pero es aun peor que estos se presenten durante el funcionamiento del sistema, talvez aspectos que no se han tomado en cuenta, inconvenientes que pueden afectar lo ya presentado como producto final, y que cambien la garantía de calidad que se aseguro; por esa razón es de vital importancia que la gestión de los riesgos se realice periódicamente a medida que avanza el proyecto de desarrollo, y no sobredimensionar de resultados en un proyecto, ya que pueden traer varios inconvenientes creando expectativas mayores de las que el grupo

de trabajo pueda desarrollar, en la que puede verse afectado la planificación, las horas de trabajo, calidad del producto.

“Es importante que durante la vida del proyecto se procure reducir las situaciones adversas que se pueden definir como riesgos y aumentar el impacto de los eventos positivos”. (Gestión de Riesgos en la Administración de Proyectos [7])

“En la Tabla 6 indica las consecuencias potenciales de riesgos (filas etiquetadas con 1) o la imposibilidad de conseguir el producto deseado (filas etiquetadas con 2). La categoría de impacto es elegida basándose en la caracterización que mejor encaja con la descripción de la tabla.” (Riesgos del Software [21])

		<b>Rendimiento</b>	<b>Soporte</b>	<b>Costo</b>	<b>Planificación Temporal</b>
<b>Catastrófico</b>	1	Dejar de cumplir los requerimientos provocaría fracaso de la misión.		Malos resultados en un aumento de costos y retrasos de la planificación temporal con cifras que superan los US\$500.000.	
	2	Degradación significativa para no poder alcanzar el rendimiento técnico.	El software no responde o no admite soporte.	Recortes financieros significativos, presupuestos excedidos.	Fecha de entrega inalcanzable.
<b>Crítico</b>	1	Dejar de cumplir los requerimientos degradaría el rendimiento del sistema hasta un punto donde el éxito de la misión es cuestionable.		Malos resultados en retrasos operativos y/o aumento de costo con valor esperado de US\$100.000 a US\$500.000.	
	2	Alguna disminución en el rendimiento técnico.	Pequeños retrasos en modificaciones de sw.	Algunos recortes de los recursos financieros, posibles excesos del presupuesto.	Posibles retrasos de la fecha de entrega.
<b>Marginal</b>	1	Dejar de cumplir los requerimientos provocaría la degradación de la misión secundaria.		Los costos, impactos y/o retrasos recuperables de la planificación temporal con un valor estimado de US\$1000 a US\$100.000.	
	2	De mínima a pequeña reducción en el rendimiento técnico.	EL soporte del software no responde.	Recursos financieros suficientes.	Planificación temporal realista, alcanzable.
<b>Despreciable</b>	1	Dejar de cumplir los requerimientos provocaría inconvenientes o impactos no operativos.		Los errores provocan impactos mínimos en el costo y/o planificación temporal con un valor esperado de menos de US\$1000.	
	2	No hay reducción en el rendimiento técnico.	Software fácil de dar soporte.	Posible superávit de presupuesto.	Fecha de entrega fácilmente alcanzable.

Tabla 6. Categoría versus componentes.

Fuente: Riesgos del Software [21]

### **3.2. PROPUESTA PARA LOGRAR LA REDUCCIÓN DE RIESGOS.**

La mayoría de metodologías ágiles poseen principios y características, que ayudan en la gestión de riesgos; en el caso de Scrum, las reuniones que se realizan para la planificación, seguimiento y presentación de los resultados de la iteración son una de las herramientas que ayudan al control de riesgos, ya que al existir constante comunicación, se puede identificar los posibles problemas de forma temprana y aplicar la solución más acertada. Scrum sin duda brinda muchas posibilidades, y beneficios, que ayudan a gestionar los posibles riesgos en los proyectos de desarrollo de software, siendo algo que no se maneja de forma independiente, ya que se experimenta durante toda la vida del proyecto, y esto no es tomado como una actividad complementaria, sino como algo que se realiza implícitamente, de forma natural y continua en cada actividad que maneja esta metodología.

La forma con la que Scrum maneja el desarrollo por medio de iteraciones, ayuda a tener un mejor control en este tipo de ambientes, que por su corta duración entre cada sprint, son de mejor manejo al momento de poder identificar anomalías y problemas.

Existen divergencias sobre quién debe ser el responsable de la gestión de riesgos, ya que podría recaer sobre el dueño del producto, el scrum master; o simplemente todo el grupo de trabajo, incluyendo el cliente; sin embargo esto debe acoplarse a las necesidades que demanden los riesgos en el proyecto, y a la disponibilidad y apertura de los miembros del grupo de trabajo.

#### **3.2.1. CONSIDERACIONES DE LA PROPUESTA.**

A continuación se detallan las consideraciones, que se deben tomar en cuenta previo a la ejecución de la propuesta:

- La siguiente propuesta puede ser aplicada en proyectos de cualquier alcance, siempre y cuando estos usen la metodología Scrum, para su desarrollo.
- Cada etapa para la gestión de riesgos, debe involucrar a todos los miembros del grupo de trabajo, incluyendo al propietario del producto o cliente. De esta manera con la colaboración y comunicación que promulga Scrum, ayudará a que los riesgos y problemas se puedan detectar, y corregir posteriormente, con la ayuda de todos los miembros del grupo de trabajo.
- Asignar un responsable para la gestión de riesgos, según las necesidades del proyecto y la mejor forma para adaptarse a esto, se puede seleccionar entre el propietario del producto y el Scrum Manager.
- Tomar los formatos y documentos que se propondrán, servirán como base y ayuda para la gestión de riesgos.
- La propuesta no podrá manejar riesgos externos al proyecto, que no puedan ser controlados por ninguna parte involucrada en el proyecto.
- Los formatos presentados para la gestión de riesgos, son elaborados por la autora de la propuesta. Los encabezados de los documentos deberán ser llenados de acuerdo a la información requerida, que corresponde a: nombre del proyecto, iteración, fecha, y versión del documento; de esta manera en el futuro podrán ser mejor identificados los diferentes documentos.

### **3.2.2. OBJETIVOS DE LA PROPUESTA.**

Durante la implementación de la propuesta y al finalizarla, se intenta conseguir algunos objetivos, que se describen a continuación:

- Lograr reducir los riesgos en los proyectos donde se implemente la propuesta, a medida que estos avancen y se familiaricen con el proceso de gestión de riesgos.

- Conseguir concientizar a las personas que vayan aplicar la propuesta, sobre la importancia de realizar la gestión de riesgos respectiva, sobre los proyectos de desarrollo de software.

### 3.2.3. PROPUESTA.

Para poder realizar el proceso de gestión de riesgos, se planteará etapas, las mismas que contendrán tareas; esto se resume en la Tabla 7, y posteriormente se detallan cada una.

ETAPAS	TAREAS
Etapa 1: Identificación de Riesgos.	Tarea 1: Obtener información de riesgos genéricos.
	Tarea 2: Obtener información de riesgos específicos del proyecto.
Etapa 2: Análisis de Riesgos.	Tarea 3: Definir grado de probabilidad de los riesgos.
	Tarea 4: Establecer consecuencias de los riesgos y nivel de criticidad en el proyecto.
	Tarea 5: Especificar la prioridad de los riesgos.
Etapa 3: Planteamiento y gestión de soluciones.	Tarea 6: Asignar nivel de gestión de riesgos.
	Tarea 7: Planteamiento de soluciones.
	Tarea 8: Ejecución de solución.
	Tarea 9: Control de la solución.
Etapa 4: Análisis de resultados.	Tarea 10: Realizar análisis de resultados acerca del grado de reducción de riesgos.
Etapa 5. Continuidad de la gestión de riesgos.	Tarea 11. Repetir el ciclo de gestión de riesgos.

Tabla 7. Resumen de etapas de la propuesta para la reducción de riesgos.

Autor: Lilián Elizabeth Arroba Medina.

**Etapa 1: Identificación de Riesgos.**

Esta etapa se deberá realizar durante las primeras fases del ciclo de Scrum, es decir, durante la generación, planificación y seguimiento del Product Backlog y el Sprint Backlog.

Dado que el Sprint Backlog se genera por cada nueva iteración a desarrollarse, la identificación de riesgos no solo será una tarea de la parte inicial del proyecto, sino será un proceso continuo y de retroalimentación durante todo el proyecto.

**Tarea 1: Obtener información de riesgos genéricos.**

Esta información será un aporte de cada miembro del grupo de trabajo, la que corresponderá a riesgos comunes que pueden experimentar cualquier tipo de proyecto, esto podrá basarse en la experiencia previa que han tenido en otros proyectos.

En este listado de riesgos, solamente se deberá indicar aquellos que sean considerados como genéricos, es decir, los que ocurren de forma común en los proyectos sin que sean afectados de manera drástica por las características propias del proyecto, ya que para estos, posteriormente se deberán brindar soluciones genéricas que se ajusten al proyecto, las que se implementarán en el momento oportuno.

Con la información identificada posteriormente se llenará los siguientes campos del Formato 1: id del riesgo y descripción.

- **ID del riesgo:** esta nomenclatura puede ser usada para manejar el riesgo de manera más rápida, es decir, al referirse a un determinado riesgo se puede utilizar su identificador, y no necesariamente la descripción completa del mismo. La forma de asignar el identificador se seleccionará de la forma que crea adecuada el grupo de trabajo, ya que se podrá iniciar con un número secuencial, o se podrá asignar alguna nomenclatura que vaya acorde con el estándar que maneje el grupo.



quienes después de verificar los posibles inconvenientes que se encuentren en el desarrollo del sistema, puedan definir a estos, como riesgos para el proyecto.

Una vez identificado los riesgos específicos se deberá llenar el Formato 2, con la información requerida para esta etapa, la cual corresponde a: id del riesgo y descripción.

- **ID del riesgo:** esta nomenclatura puede ser usada para manejar el riesgo de manera más rápida, es decir, al referirse a un determinado riesgo, se puede utilizar su identificador, y no necesariamente la descripción completa del mismo. La forma de asignar el identificador se seleccionará de la forma que crea más adecuada el grupo de trabajo, ya que se podrá iniciar con un número secuencial, o se podrá asignar alguna nomenclatura que vaya acorde con el estándar que maneje el grupo.
- **Descripción:** en este campo se detallará el riesgo, de debe procurar usar las palabras correctas que describan con precisión el riesgo, evitando palabras innecesarias que no contribuyan a entender de mejor manera el problema y que solo extienden el contenido de este campo.

**DOCUMENTO PARA IDENTIFICACIÓN DE RIESGOS****RIESGOS ESPECÍFICOS**

NOMBRE DEL PROYECTO:

ITERACIÓN:

FECHA:

VERSIÓN DEL DOCUMENTO:

ID RIESGO	DESCRIPCIÓN	CAUSAS	% DE OCURRENCIA	NIVEL DE CRITICIDAD	CONSECUENCIAS

---

 FIRMA DE RESPONSABILIDAD

Formato 2. Documento de identificación de riesgos específicos.

**Etapas 2: Análisis de Riesgos.**

Esta etapa se desarrollará durante dos fases del ciclo de Scrum las cuales son: definición de tareas y responsabilidades, y seguimiento de la iteración. Se recomienda que la tarea 3 se realice solamente en la primera fase mencionada del proceso de Scrum, la tarea 4 se realizará en las dos fases de Scrum, compartiendo su desarrollo según el criterio del grupo de trabajo, y la tarea 5 solamente se desarrollará en la segunda fase mencionada de la metodología.

**Tarea 3: Definir grado de probabilidad de los riesgos.**

Una vez identificados los riesgos del proyecto, tanto genéricos como específicos, se deberá asignar un porcentaje de probabilidad para que estos riesgos se presenten, en

el caso de que estos no se hayan generado aún; y en el caso de aquellos que ya se han presentado, se definirá la probabilidad con que estos puedan repetirse. Para poder definir el porcentaje, es necesario que se analicen las razones de origen de los riesgos, esto ayudará a tener una mejor visión de cualquier tipo de riesgo. Este valor deberá ser un acuerdo de todos los miembros del proyecto, para esto se puede usar la estimación del poker, utilizada también para asignar tiempos de ejecución de las tareas en la planificación de los sprints.

Posteriormente en los Formatos 1 y 2, que corresponden a los riesgos genéricos y específicos, respectivamente, se llenará los siguientes campos: causas y porcentaje de ocurrencia.

- **Causas:** en este campo se deberá especificar las principales causas para la generación de un determinado riesgo, evitando información innecesaria que no aporte valor.
- **Porcentaje de ocurrencia:** este número se calculará en base a la experiencia y conocimiento de los integrantes del grupo de trabajo, el cual a la vez será un promedio de los porcentajes propuestos por cada integrante, con la técnica de estimación de poker.

#### **Tarea 4: Establecer consecuencias de los riesgos y nivel de criticidad en el proyecto.**

Para cada riesgo identificado en las tareas anteriores, se debe establecer las posibles consecuencias o las ya generadas, que afectarán el proyecto, y que tan críticas serán estas para garantizar el éxito. Esta información deberá ser un aporte y a la vez un acuerdo de todos los miembros del proyecto, ya que mientras más datos se puedan obtener sobre el riesgo, se logrará definir de forma más precisa el nivel de criticidad.

Posteriormente se deberá llenar en los Formatos 1 y 2 los siguientes campos: nivel de criticidad y consecuencias.

- **Nivel de criticidad:** este campo se debe seleccionar entre las siguientes posibilidades:
  - Despreciable.
  - Marginal.
  - Critico.
  - Catastrófico.
- **Consecuencias:** se deberá analizar, y posteriormente detallar los efectos o consecuencias asociadas al riesgo, con esto se puede entender el grado de impacto sobre el proyecto.

#### **Tarea 5: Especificar la prioridad de los riesgos.**

Con el nivel de criticidad ya definido en la tarea anterior, se podrá especificar de mejor manera la prioridad de atención que se dará a cada riesgo, de esta manera poder atender aquellos que puedan perjudicar mayormente al proyecto.

Con la información analizada se llenara los siguientes campos en el Formato 3:

- **ID riesgo:** este campo corresponderá a la nomenclatura utilizada en los Formatos 1 y 2 de identificación de riesgos.
- **Descripción:** este campo deberá contener la misma información que se uso en los Formatos 1 y 2 de identificación de riesgos, a la vez este campo puede ser opcional, ya que se puede hacer referencia al riesgo solo con su identificador.
- **Nivel de prioridad:** este campo deberá detallar los tres posibles niveles que se manejarán para esto, y que se detallan a continuación:
  - Alto: los riesgos que se manejen en este nivel, serán aquellos que necesiten atención urgente, es decir de forma rápida, pero sin olvidar que se deberá brindar la solución optima.
  - Medio: los riesgos que se manejen en este nivel, serán aquellos que no necesitan atención de forma inmediata.

- Bajo: los riesgos que se manejen en este nivel, serán aquellos que para ejecutar su solución se puede tomar mayor tiempo, ya que no afectan radicalmente al proyecto.
- **Orden de atención:** de acuerdo al nivel de prioridad asignado a los riesgos se asignará el orden en el cuál serán solucionados, es importante indicar que pueden existir casos en los que la solución de varios riesgos sea simultánea, por lo que deberán tener el mismo orden de atención.
- **Observaciones:** este campo será opcional, en el cual se puede acotar información adicional, que sea de ayuda para entender la información ingresada en el formato.

Para el Formato 3 que será usado en esta tarea, el grupo debe decidir cómo lo generará, es decir si se crea un documento por cada categoría de riesgo (genéricos o específicos), o un documento que agrupe las dos categorías.

<b>DOCUMENTO PARA PRIORIZACIÓN DE RIESGOS</b>				
<b>RIESGOS GENÉRICOS Y/O ESPECÍFICOS</b>				
<b>NOMBRE DEL PROYECTO:</b>			<b>ITERACIÓN:</b>	
<b>FECHA:</b>				
<b>VERSIÓN DEL DOCUMENTO:</b>				
ID RIESGO	DESCRIPCIÓN	NIVEL DE PRIORIDAD	ORDEN DE ATENCIÓN	OBSERVACIONES

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

Formato 3. Documento para priorización de riesgos.

**Etapa 3: Planteamiento y gestión de soluciones.**

Esta etapa se desarrollará durante la fase del ciclo de Scrum, seguimiento de la iteración, que es en donde se tendrá mayor acción para lograr el objetivo de la iteración, y por lo tanto se podrá tener un mejor panorama para la solución de los riesgos en gestión.

**Tarea 6: Asignar nivel de gestión de riesgos.**

Para el formato que se utilizará en esta etapa, se llevará el orden de acuerdo al campo orden de atención del formato 3, en esta etapa se irá analizando la solución de cada riesgo según el orden seleccionado anteriormente.

Además de los campos ya utilizados anteriormente en otros formatos, id riesgo y descripción, se deberá ingresar en el Formato 4 el siguiente campo:

- **Nivel de gestión:** Dependiendo del momento en el que sea detectado el riesgo, se podrá definir qué tipo de acciones se realizarán para la respectiva gestión, entre estos tipos o niveles podemos definir los siguientes: (Gestión de Riesgos en Proyectos Software [8])
  - Control de crisis: intentar controlar los riesgos cuando ya se han convertido en problemas.
  - Arreglar cada error: reaccionar rápidamente cuando ya se ha producido el riesgo.
  - Mitigación de riesgos: planificar con antelación el tiempo que necesitaría para cubrir riesgos en caso de que ocurran, pero sin intentar eliminarlos inicialmente.
  - Prevención: crear y ejecutar, como parte del proyecto software, un plan para identificar riesgos y evitar que se conviertan en problemas.
  - Eliminación de causas principales: identificar y eliminar los factores que pueden provocar la presencia de algún tipo de riesgo.



- **Solución:** este campo debe detallar de la manera precisa la solución seleccionada por el grupo, para que esta pueda ser entendida de la mejor forma al momento de su ejecución.
- **Tiempo estimado:** se puede utilizar la técnica de estimación de poker, y de esta manera llegar a un acuerdo del tiempo que se empleará en la solución del riesgo.
- **Responsable:** en este campo se indicará la persona o personas que serán las responsables de la ejecución de la solución, siendo óptimo que se seleccione a la persona que se encuentre familiarizada con el tema que involucra el riesgo.

#### **Tarea 8: Ejecución de solución.**

En esta tarea una vez definidos todos los aspectos relacionados con la solución, se deberá proceder con la ejecución de la misma, bajo los parámetros ya detallados y acordados anteriormente.

#### **Tarea 9: Control de la solución.**

En el tiempo que tomará la solución se debe llevar un control sobre la misma, indicando cierta información que ayuda con la respectiva revisión.

Para el Formato 5 usado en esta tarea, además de los campos ya utilizados anteriormente, id riesgo y descripción, se debe ingresar los campos: estado de la solución y observaciones:

- **Estado de la solución:** para llenar este campo se puede usar términos que se acoplen de mejor manera a la forma de trabajar en el grupo. A continuación se detallan algunos ejemplos de la forma de manejar este campo:
  - Porcentajes: se podrá indicar el porcentaje de la solución ya ejecutado para verificar si esta avanza de acuerdo a lo establecido.
  - Estados: se podrá indicar el estado en el que se encuentra la solución: pendiente, en curso y solucionado; estos estados también son los

recomendados para el control de las tareas que involucran los requerimientos del proyecto.

- **Observaciones:** en este campo se podrá indicar si han existido cambios en lo establecido, u otra información que sea de ayuda para el control.

DOCUMENTO PARA CONTROL DE SOLUCIÓN DE RIESGOS			
RIESGOS GENÉRICOS Y/O ESPECÍFICOS			
NOMBRE DEL PROYECTO:		ITERACIÓN:	
ITERACIÓN:			
FECHA:			
VERSIÓN DEL DOCUMENTO:			
ID RIESGO	DESCRIPCIÓN	ESTADO DE LA SOLUCIÓN	OBSERVACIONES
<hr style="width: 20%; margin: 0 auto;"/> FIRMA DE RESPONSABILIDAD			

Formato 5. Documento para control de solución de riesgos.

Para el control de las soluciones de los riesgos se podrá utilizar adicionalmente un elemento familiar en el proyecto, como es el pizarrón de seguimiento de Scrum, en el cual los riesgos se irán moviendo de acuerdo a cómo evoluciona la solución de los mismos. Esto se representa en la Figura 8.

RIESGO	Pendiente	En curso	Solucionado
Descripción	Observación Responsable Duración		
Descripción	Observación Responsable Duración	Observación Responsable Duración	
Descripción	Observación Responsable Duración		Observación Responsable Duración

Figura 8. Representación del avance de gestión de riesgos en un pizarrón.

Autor: Lilián Elizabeth Arroba Medina

#### **Etapas 4: Análisis de resultados.**

Esta etapa se desarrollará durante la fase del ciclo de Scrum, que corresponde a la presentación de la iteración, ya que esta consta con el periodo para poder realizar la pertinente retrospectiva de la iteración y a cada elemento de la misma, lo que incluye la gestión de los riesgos presentados para dicha iteración.

#### **Tarea 10: Realizar análisis de resultados acerca del grado de reducción de riesgos.**

Esta tarea es de importancia para poder cuantificar los niveles de reducción de riesgos, los cuales se podrán evidenciar de mejor manera a medida que el proyecto vaya avanzando, y así comprobar que una gestión oportuna sobre los diferentes riesgos, hará que aquellos que se presentaron en etapas iniciales, dejen de ser una amenaza para el proyecto a medida que este ha ido avanzando.

Para este análisis, se lo hará en dos partes: la primera se enfocará en definir el porcentaje de riesgos solventados, y la segunda en aquellos que no se han solucionado.

Se deberán tomar solamente aquellos riesgos que se planificó solucionarlos al final de la iteración, ya que podrían existir algunos que se planeo solventarlos en un periodo superior al de la finalización de la iteración, los que no deben ser analizados aun, sin embargo riesgos que se encuentren en este caso no deberían existir o al menos ser mínimos, ya que los generados en cada iteración deben ser solucionados al finalizar la misma, de otra manera esto significaría que el éxito del cumplimiento del objetivo del sprint se podría haber afectado.

Para la primera parte se realizará un análisis de los riesgos gestionados, para de esta manera cuantificar el porcentaje de riesgos que se solucionaron según lo planificado. Podrán incluirse aquellos riesgos para los cuales se plantearon soluciones para después de la finalización del sprint, pero que sin embargo se resolvieron antes de lo planificado, y que se encuentran en el periodo previo a la culminación de la iteración.

El siguiente documento puede ser generado una sola vez para todo el proyecto, ya que en este análisis se indicará el número de sprint analizado. En el Formato 6 se deberán llenar los siguientes campos:

- **Fecha:** se debe indicar la fecha en la que se realiza el análisis.
- **N. Sprint:** en este campo se indicará el número de sprint en el que se analiza los riesgos.
- **Número de riesgos en análisis:** este campo indicará el número de riesgos que se gestionaron dentro del periodo de la iteración.
- **Porcentaje de riesgos solucionado:** este campo llevará el porcentaje de riesgos que fueron solucionados exitosamente dentro del periodo de la iteración.
- **Porcentaje de riesgos no solucionados:** este campo llevará el porcentaje de riesgos que no se han solucionado hasta la fecha de finalización del sprint.
- **Observaciones:** en este campo se indicará alguna información adicional, que sea de ayuda y pueda aportar valor al análisis.



Una vez identificado el listado de riesgos gestionados hasta el final del sprint, se deberá tomar solamente aquellos que no fueron solucionados, a pesar de que se planificó solventarlos al final de la iteración, de estos se deberán analizar las razones para no haberlos solucionado según lo planificado, y posteriormente llenar los campos del Formato 7:

- **ID Riesgo:** este campo corresponderá a la nomenclatura utilizada en los formatos anteriores.
- **Descripción:** este campo detallará la descripción usada en otros formatos para identificar el riesgo.
- **Razones para no finalizar solución:** se detallará las principales razones para no haber completado la solución planteada para el riesgo.
- **Porcentaje de solución realizada:** se debe indicar en este campo el porcentaje ya realizado de la solución.
- **Se requiere cambio de solución:** según el porcentaje indicado en el campo anterior se podrá decidir si el riesgo en análisis requiere realizar alguna modificación total o parcial de la solución, de ser el caso que si lo requiera, se deberá marcar este campo, para que posteriormente el riesgo sea analizado, y que forme parte nuevamente de los Formatos 3, 4 y 5 en conjunto con los nuevos riesgos generados para la siguiente iteración. Para el Formato 3 se recomienda que estos riesgos aun no solucionados se encuentren primeros en orden de solución, ya que estos ya deberían haber sido resueltos.



## **CAPÍTULO IV.**

### **APLICACIÓN DE LA PROPUESTA UTILIZANDO SCRUM.**

#### **4.1. CASO DE ESTUDIO.**

##### **4.1.1. DESCRIPCIÓN DE LA EMPRESA.**

La empresa con la que se trabajará para la aplicación de la propuesta será TV Cable, que es una de las empresas pioneras en el Ecuador, en el área de servicios de Telecomunicaciones, que brinda a sus clientes productos de calidad.

El área con la que se trabajará en la ejecución de la propuesta, es la que se encuentra encargada del desarrollo e implementación de los diferentes Sistemas de Gestión de la empresa.

Tanto la empresa como el área mencionada han brindado la colaboración necesaria para ejecutar la propuesta durante el desarrollo de una de sus aplicaciones.

##### **4.1.2. DESCRIPCIÓN DEL SISTEMA.**

El sistema sobre el cual se aplicará la propuesta es llamado Titan, esta es una aplicación que se desenvuelve en ambiente web, solo accesible a usuarios permitidos en la intranet de la empresa.

Titan es un sistema que permite la gestión de los clientes, y además maneja la comunicación entre las plataformas que brindan los diferentes servicios y el equipo físico que emite la señal.

El desarrollo de este sistema maneja de forma muy adecuada al concepto de manifiesto ágil, siendo esta una de las razones por las que se ha seleccionado el desarrollo de Titan para la respectiva aplicación de la propuesta de reducción de riesgos basada en Scrum, siendo esta la metodología que se aplica para el desarrollo del sistema en mención.

Titan es desarrollado por iteraciones que manejan periodos de tiempo de dos semanas, son iteraciones cortas ya que cada una de estas se encuentran manejando el criterio de mejora continua, para permitir a cada proceso que maneja el sistema actualmente, tener un mejor nivel de accesibilidad hacia el usuario interno y una funcionalidad más amplia que abarque las necesidades de las áreas involucradas.

#### **4.1.3. OBJETIVOS DE LA APLICACIÓN DE LA PROPUESTA.**

El desarrollo del sistema Titan es un proyecto que ha manejado como cualquier otro tipo de proyecto, diferentes tipos de riesgos a lo largo de su ciclo de vida, implementando la propuesta se desea lograr algunos objetivos que vayan marcando pautas para lograr la reducción de riesgos en el proyecto.

Entre los objetivos enmarcados en la propuesta, se van a tomar los siguientes:

- Lograr un porcentaje de reducción de riesgos que vaya incrementando a medida que el proyecto avance.
- Incentivar al grupo de trabajo a realizar un proceso continuo de gestión de riesgos que se puedan presentar en el ciclo de vida del sistema Titan.
- Mejorar la identificación de riesgos para poder atenderlos oportunamente, y de esta manera en muchos casos minimizar el impacto en el proyecto
- Conseguir en los miembros del grupo de trabajo, familiarización con la propuesta de reducción de riesgos para poder implementarla progresivamente durante todas las iteraciones.

## **4.2. APLICACIÓN DE LA PROPUESTA Y ANÁLISIS DE RESULTADOS.**

### **4.2.1. APLICACIÓN DE LA PROPUESTA.**

Para la aplicación de la propuesta se analizarán 3 iteraciones, en las mismas se identificarán los riesgos presentados y se gestionarán de acuerdo a la propuesta planteada.

Los siguientes documentos son generados por la autora de la propuesta, en base a la información recopilada durante la gestión de riesgos del sistema antes indicado, siguiendo los formatos de la propuesta presentada en el capítulo 3.

Para identificar a los riesgos se uso la nomenclatura G para los genéricos, y E para los específicos, con la numeración secuencial correspondiente.

A continuación se muestran los documentos generados para cada iteración, en cada una de las tareas de la propuesta.

### Iteración 1

Para la primera iteración se realizó el proceso de gestión de riesgos propuesto, para lo cual se generaron los Documentos 1 y 2, los que contienen información recopilada y analizada en las tareas 1 hasta la 4, correspondientes a las etapas 1 y 2 de la propuesta. En estas tareas se identificaron 2 riesgos genéricos y 4 específicos, los que se describen a continuación:

<b>DOCUMENTO PARA IDENTIFICACIÓN DE RIESGOS</b>					
<b>RIESGOS GENÉRICOS</b>			<b>ITERACIÓN: 1</b>		
<b>NOMBRE DEL PROYECTO: Titan</b>					
<b>FECHA: 2010 – 12 – 23</b>					
<b>VERSION DEL DOCUMENTO: 1.0</b>					
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>CAUSAS</b>	<b>% DE OCURRENCIA</b>	<b>NIVEL DE CRITICIDAD</b>	<b>CONSECUENCIAS</b>
G1	Tiempos de planificación incorrectamente programados.	<ul style="list-style-type: none"> <li>Falta de análisis al plantear fechas de programación.</li> <li>No contemplar correctamente el alcance de los requisitos.</li> <li>Plantear fechas sin analizar el recurso humano disponible.</li> </ul>	30%	Marginal.	<ul style="list-style-type: none"> <li>Uso de más recursos a lo programado.</li> <li>Retraso en las fechas límites de entrega de los productos.</li> <li>Esfuerzo excesivo por alcanzar fechas planificados.</li> </ul>
G2	Falta de comunicación entre los miembros del grupo de desarrollo.	<ul style="list-style-type: none"> <li>Constante divergencia de opiniones al tomar decisiones que afectan a todo el grupo.</li> <li>No contar con un ambiente agradable para fomentar la comunicación constante.</li> <li>Fomentar solo la comunicación en momentos de problemas para el proyecto.</li> </ul>	30%	Crítico.	<ul style="list-style-type: none"> <li>No generar constante retroalimentación en el proyecto.</li> <li>No lograr el objetivo de la iteración.</li> <li>Empeorar el ambiente de trabajo.</li> </ul>

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

**DOCUMENTO PARA IDENTIFICACIÓN DE RIESGOS****RIESGOS ESPECÍFICOS****NOMBRE DEL PROYECTO:** Titan**ITERACIÓN:** 1**FECHA:** 2010 – 12 – 23**VERSIÓN DEL DOCUMENTO:** 1.0

<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>CAUSAS</b>	<b>% DE OCURRENCIA</b>	<b>NIVEL DE CRITICIDAD</b>	<b>CONSECUENCIAS</b>
E1	Cambio de responsable en tareas asignadas.	<ul style="list-style-type: none"> <li>• Distribución de responsabilidades no equitativa.</li> <li>• No considerar habilidades de los miembros del grupo de trabajo al asignar tareas.</li> </ul>	10%	Marginal.	<ul style="list-style-type: none"> <li>• Retrasos en las fechas de entrega de los productos.</li> <li>• Demora en acoplarse el nuevo responsable de la tarea.</li> </ul>
E2	El cliente no asiste a citas pactadas con anterioridad.	<ul style="list-style-type: none"> <li>• El cliente no contempló la reunión en su cronograma.</li> <li>• No brindar la importancia necesaria al intercambio de información que se realizaría en dicha cita.</li> </ul>	5%	Crítico.	<ul style="list-style-type: none"> <li>• No poseer la información necesaria para desarrollar el requerimiento planificado, y bajo las características que el cliente requiere.</li> </ul>
E3	Dependencia de otras aplicaciones para continuar con el desarrollo.	<ul style="list-style-type: none"> <li>• Se requiere configuraciones previas en otros sistemas, para poder brindar la funcionalidad requerida al proceso en desarrollo.</li> </ul>	30%	Crítico.	<ul style="list-style-type: none"> <li>• No poseer total autonomía para poder realizar el desarrollo de los procesos, impidiendo la verificación y pruebas respectivas de la funcionalidad.</li> </ul>
E4	Poca apertura por parte de los usuarios para manejar el sistema.	<ul style="list-style-type: none"> <li>• No proporcionar verdadera importancia a la ayuda que brinda el sistema.</li> <li>• Poca colaboración e interés en la capacitación respectiva.</li> </ul>	40%	Despreciable.	<ul style="list-style-type: none"> <li>• Retrasos en el uso del sistema, ya que se debe capacitar nuevamente.</li> <li>• Los usuarios convierten el soporte de la aplicación en una ayuda demasiado constante.</li> </ul>

---

 FIRMA DE RESPONSABILIDAD

En la tarea 5 se procedió a generar un documento que agrupe tanto a riesgos genéricos como específicos, en el cual se analizó la priorización de atención que se dio a los riesgos, esta información se describe a continuación en el Documento 3:

<b>DOCUMENTO PARA PRIORIZACIÓN DE RIESGOS</b>				
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>				
<b>NOMBRE DEL PROYECTO: Titan</b>			<b>ITERACIÓN: 1</b>	
<b>FECHA: 2010 – 12 – 27</b>				
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>				
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>NIVEL DE PRIORIDAD</b>	<b>ORDEN DE ATENCIÓN</b>	<b>OBSERVACIONES</b>
G1	Tiempos de planificación incorrectamente programados.	Alto.	1	
G2	Falta de comunicación entre los miembros del grupo de desarrollo.	Bajo.	5	
E1	Cambio de responsable en tareas asignadas.	Alto.	2	
E2	El cliente no asiste a citas pactadas con anterioridad.	Alto.	3	
E3	Dependencia de otras aplicaciones para continuar con el desarrollo.	Medio.	4	
E4	Poca apertura por parte de los usuarios para manejar el sistema.	Bajo.	6	

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

En la etapa 3 correspondiente al planteamiento y gestión de soluciones, se ejecutaron las tareas 6 y 7; de las cuales se generó el Documento 4, descrito a continuación, el cual contiene la información necesaria para realizar la tarea 8.

<b>DOCUMENTO PARA GESTIÓN DE RIESGOS</b>					
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>					
<b>NOMBRE DEL PROYECTO: Titan      ITERACIÓN: 1</b>					
<b>FECHA: 2010 – 12 – 28</b>					
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>					
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>NIVEL DE GESTIÓN</b>	<b>SOLUCIÓN</b>	<b>TIEMPO ESTIMADO</b>	<b>RESPONSABLE</b>
G1	Tiempos de planificación incorrectamente programados.	Mitigación de riesgos.	A pesar de que en la metodología Scrum no es imprescindible cumplir estrictamente la planificación, sin embargo si se tomará mayor tiempo para analizar los tiempos de ejecución de tareas lo mas aproximados posibles.	2 horas.	Gerente / Jefe de área de proyectos.
G2	Falta de comunicación entre los miembros del grupo de desarrollo.	Eliminación de causas principales.	Scrum incentiva la constante comunicación constante en los miembros del grupo de trabajo, por lo que se fomentará mayormente esto, ya que es de vital importancia para reducir las posibilidades de ocurrencia de este tipo de riesgo.	6 horas.	Grupo de desarrollo.
E1	Cambio de responsable en tareas asignadas.	Prevención.	Realizar la distribución equitativa de tareas, tomando en cuenta habilidades y destrezas.	2 horas.	Gerente / Jefe de área de proyectos.
E2	El cliente no asiste a citas pactadas con anterioridad.	Eliminación de causas principales.	Incentivar el compromiso en el cliente, para que pueda brindar la información adecuada de manera oportuna al grupo de desarrollo.	3 horas.	Cliente.
E3	Dependencia de otras aplicaciones para continuar con el desarrollo.	Prevención.	Lograr que las configuraciones necesarias en otros sistemas se encuentren realizadas para continuar el desarrollo, acordando la colaboración de las personas involucradas.	3 horas.	Grupo de desarrollo.
E4	Poca apertura por parte de los usuarios para manejar el sistema.	Eliminación de causas principales.	Demostrar los beneficios del sistema para los usuarios, de manera que se comprometan para poder aprender con apertura su funcionamiento.	6 horas.	Vicepresidente / Gerente de área.

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

En la tarea 9 perteneciente al control de la solución, se generó el Documento 5, descrito a continuación:

**DOCUMENTO PARA CONTROL DE SOLUCIÓN DE RIESGOS**

**RIESGOS GENÉRICOS Y ESPECÍFICOS**

**NOMBRE DEL PROYECTO:** Titan

**ITERACIÓN:** 1

**FECHA:** 2010 – 12 – 30

**VERSIÓN DEL DOCUMENTO:** 1.0

<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>ESTADO DE LA SOLUCIÓN</b>	<b>OBSERVACIONES</b>
G1	Tiempos de planificación incorrectamente programados.	100%	
G2	Falta de comunicación entre los miembros del grupo de desarrollo.	30%	
E1	Cambio de responsable en tareas asignadas.	100%	
E2	El cliente no asiste a citas pactadas con anterioridad.	70%	
E3	Dependencia de otras aplicaciones para continuar con el desarrollo.	50%	
E4	Poca apertura por parte de los usuarios para manejar el sistema.	40%	

---

FIRMA DE RESPONSABILIDAD

Al finalizar el análisis de la gestión de riesgos para la primera iteración, se generó el Documento 6 que corresponde a la tarea 10 de la etapa 4 de la propuesta, que pertenece al análisis de resultados. Este documento fue generado una sola vez para las tres iteraciones en análisis.

<b>DOCUMENTO ANÁLISIS DE GESTIÓN DE RIESGOS</b>					
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>			<b>ITERACIÓN: 1</b>		
<b>NOMBRE DEL PROYECTO: Titan</b>					
<b>FECHA: 2011 - 01 - 06</b>					
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>					
<b>FECHA</b>	<b>N. SPRINT</b>	<b>NÚMERO DE RIESGOS EN ANÁLISIS</b>	<b>PORCENTAJE RIESGOS SOLUCIONADOS</b>	<b>PORCENTAJE RIESGOS NO SOLUCIONADOS</b>	<b>OBSERVACIONES</b>
2011 - 01 - 06	1	6	100%	0	
2011 - 01 - 25	2	5	100%	0	
2011 - 02 - 06	3	4	100%	0	

\_\_\_\_\_  
FIRMA DE RESPONSABILIDAD

## Iteración 2

En la iteración número 2 se continúa con la tarea 11, que corresponde a la repetición del ciclo de gestión de riesgos. Para este sprint, se generaron los Documentos 7 y 8, correspondientes a las tareas 1,2 3, y 4, de las etapas 1 y 2 de la propuesta, que corresponden a la identificación y análisis de riesgos. En este sprint se identificaron 2 riesgos genéricos y 3 específicos, los que se describen a continuación:

<b>DOCUMENTO PARA IDENTIFICACIÓN DE RIESGOS</b>					
<b>RIESGOS GENÉRICOS</b>			<b>ITERACIÓN: 2</b>		
<b>NOMBRE DEL PROYECTO:</b> Titan					
<b>FECHA:</b> 2011 – 01 – 12					
<b>VERSIÓN DEL DOCUMENTO:</b> 1.0					
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>CAUSAS</b>	<b>% DE OCURRENCIA</b>	<b>NIVEL DE CRITICIDAD</b>	<b>CONSECUENCIAS</b>
G3	No involucrar a todas las áreas que deberían estar implicadas en el proceso.	<ul style="list-style-type: none"> <li>Falta de información para conocer y analizar qué áreas se deben involucrar para cada proceso.</li> <li>Poca conocimiento por los miembros del negocio del funcionamiento de algún proceso y como poder optimizarlo.</li> </ul>	20%	Marginal.	<ul style="list-style-type: none"> <li>Usar más tiempo a lo esperado en analizar los procesos.</li> <li>No lograr de la manera esperada la finalización del objetivo de la iteración.</li> </ul>
G4	Falta de compromiso por parte del cliente.	<ul style="list-style-type: none"> <li>El cliente no se siente parte del grupo de trabajo.</li> <li>El cliente no tiene claras las responsabilidades que debe cumplir en el proceso de desarrollo.</li> </ul>	15%	Crítico.	<ul style="list-style-type: none"> <li>Crear constantes divergencias de opiniones entre el cliente y el resto del grupo de trabajo.</li> <li>No lograr un intercambio óptimo de información para el desarrollo del proyecto.</li> </ul>

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

**DOCUMENTO PARA IDENTIFICACIÓN DE RIESGOS****RIESGOS ESPECÍFICOS****NOMBRE DEL PROYECTO:** Titan**ITERACIÓN:** 2**FECHA:** 2011 – 01 – 12**VERSIÓN DEL DOCUMENTO:** 1.0

<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>CAUSAS</b>	<b>% DE OCURRENCIA</b>	<b>NIVEL DE CRITICIDAD</b>	<b>CONSECUENCIAS</b>
E5	Poco material de apoyo para guiar a los usuarios finales en el funcionamiento de los procesos.	<ul style="list-style-type: none"> <li>Manuales con información básica que no contempla todos los posibles escenarios para algunos procesos.</li> <li>Cambios que no han sido registrados en los manuales.</li> </ul>	50%	Marginal.	<ul style="list-style-type: none"> <li>Falta de interés en los usuarios por aprender el funcionamiento del sistema.</li> <li>Utilización incorrecta de los procesos en el sistema.</li> </ul>
E6	Falta de comunicación a tiempo de decisiones gerenciales para establecer nuevos procedimientos de trabajo.	<ul style="list-style-type: none"> <li>Excesivas especulaciones de cambios por parte de las personas del área.</li> <li>Altas o bajas expectativas por parte de las personas respecto a una decisión gerencial.</li> </ul>	40%	Crítico.	<ul style="list-style-type: none"> <li>Divagación de novedades falsas que afectan en el comportamiento de las personas.</li> <li>Baja de rendimiento en el desempeño de las actividades.</li> </ul>
E7	Cliente espera producto con funcionalidad que no se acordó inicialmente.	<ul style="list-style-type: none"> <li>Cliente no tiene claro el alcance de la funcionalidad acordada.</li> <li>Cliente no contempla las posibles soluciones reales que brindará sistema al negocio.</li> </ul>	15%	Marginal.	<ul style="list-style-type: none"> <li>Perdida de colaboración por parte del cliente en el proyecto.</li> </ul>

---

 FIRMA DE RESPONSABILIDAD

Se procedió con la tarea 5, en la cual se generó el Documento 9, agrupando tanto a riesgos genéricos como específicos, analizando la priorización de atención que se dio a los riesgos, esta información se describe a continuación en el documento 9:

<b>DOCUMENTO PARA PRIORIZACIÓN DE RIESGOS</b>				
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>		<b>ITERACIÓN: 2</b>		
<b>NOMBRE DEL PROYECTO: Titan</b>				
<b>FECHA: 2011 – 01 – 13</b>				
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>				
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>NIVEL DE PRIORIDAD</b>	<b>ORDEN DE ATENCIÓN</b>	<b>OBSERVACIONES</b>
G3	No involucrar a todas las áreas que deberían estar implicadas en el proceso.	Alto.	1	
G4	Falta de compromiso por parte del cliente.	Alto.	3	
E5	Poco material de apoyo para guiar a los usuarios finales en el funcionamiento de los procesos.	Bajo.	5	
E6	Falta de comunicación a tiempo de decisiones gerenciales para establecer nuevos procedimientos de trabajo.	Alto.	2	
E7	Cliente espera producto con funcionalidad que no se acordó inicialmente.	Medio.	4	

\_\_\_\_\_  
FIRMA DE RESPONSABILIDAD

En la etapa 3 correspondiente al planteamiento y gestión de soluciones, se ejecutaron las tareas 6 y 7; generando el documento 10, con información que se describe a continuación, la misma que es necesaria para realizar la tarea 8.

<b>DOCUMENTO PARA GESTIÓN DE RIESGOS</b>					
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>					
<b>NOMBRE DEL PROYECTO: Titan</b>			<b>ITERACIÓN: 2</b>		
<b>FECHA: 2011 – 01 – 14</b>					
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>					
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>NIVEL DE GESTIÓN</b>	<b>SOLUCIÓN</b>	<b>TIEMPO ESTIMADO</b>	<b>RESPONSABLE</b>
G3	No involucrar a todas las áreas que deberían estar implicadas en el proceso.	Mitigación de riesgos.	Analizar los procesos a ser desarrollado con las partes involucradas, después de la primera revisión, para comprobar que no se ha omitido a ninguna área importante.	2 horas.	Gerente / Jefe de área de proyectos.
G4	Falta de compromiso por parte del cliente.	Mitigación de riesgos.	Crear en el cliente el sentimiento de ser parte del grupo, y que no es alguien externo, al que solo se acude para recolectar información, sino que es un elemento importante para cada etapa del proyecto y las decisiones referentes a este.	2 horas.	Vicepresidente / Gerente de área.
E5	Poco material de apoyo para guiar a los usuarios finales en el funcionamiento de los procesos.	Control de crisis.	Elaborar manuales de usuario que contemplen todos los posibles escenarios del proceso, los mismos que deben ser accesibles al usuario, y actualizados cuando sea oportuno.	5 horas.	Gerente / Jefe de área de proyectos.
E6	Falta de comunicación a tiempo de decisiones gerenciales para establecer nuevos procedimientos de trabajo.	Arreglar cada error.	Comunicar permanentemente posibles cambios a procesos y actividades de colaboradores a todos los miembros del equipo y de esta manera no generar falsas expectativas por sus miembros.	5 horas.	Gerente / Jefe de área de proyectos - Vicepresidente / Gerente de área.
E7	Cliente espera producto con funcionalidad que no se acordó inicialmente.	Eliminación de causas principales.	Redactar lo acordado en un documento donde se encuentre clara la funcionalidad a desarrollar, y comprobar con el cliente que el tema se encuentra totalmente entendido.	4 horas.	Vicepresidente / Gerente de área.

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

Se ejecutó la tarea 9 que corresponde al control de la solución, recopilando la información necesaria para la generación del Documento 11, descrito a continuación:

<b>DOCUMENTO PARA CONTROL DE SOLUCIÓN DE RIESGOS</b>			
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>		<b>ITERACIÓN: 2</b>	
<b>NOMBRE DEL PROYECTO: Titan</b>			
<b>FECHA: 2011 – 01 – 17</b>			
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>			
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>ESTADO DE LA SOLUCIÓN</b>	<b>OBSERVACIONES</b>
G3	No involucrar a todas las áreas que deberían estar implicadas en el proceso.	80%	
G4	Falta de compromiso por parte del cliente.	75%	
E5	Poco material de apoyo para guiar a los usuarios finales en el funcionamiento de los procesos.	40%	
E6	Falta de comunicación a tiempo de decisiones gerenciales para establecer nuevos procedimientos de trabajo.	70%	
E7	Cliente espera producto con funcionalidad que no se acordó inicialmente.	40%	

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

Para finalizar el análisis de la segunda iteración se realiza la tarea 10 de la etapa 4 de la propuesta, llenando la información en el Documento 6 que se generó anteriormente para el análisis de resultados de la gestión hecha a todos los riesgos generados en el proyecto.

### Iteración 3

El proceso se finalizó con el análisis de la tercera iteración, para la cual se ejecuta la tarea 11 nuevamente, que indica la repetición del proceso presentado para la gestión de riesgos. Para esta iteración se inicia con la ejecución de las tareas 1, 2, 3, 4 correspondientes a las etapas 1 y 2, en las que se realiza la identificación y análisis de riesgos; para lo cual se generaron los Documentos 12 y 13, encontrando 1 riesgo genérico y 2 específicos; los que se describen a continuación:

<b>DOCUMENTO PARA IDENTIFICACIÓN DE RIESGOS</b>					
<b>RIESGOS GENÉRICOS</b>			<b>ITERACIÓN: 3</b>		
<b>NOMBRE DEL PROYECTO: Titan</b>					
<b>FECHA: 2011- 01 - 31</b>					
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>					
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>CAUSAS</b>	<b>% DE OCURRENCIA</b>	<b>NIVEL DE CRITICIDAD</b>	<b>CONSECUENCIAS</b>
G5	Asignación incorrecta de responsabilidades.	<ul style="list-style-type: none"> <li>Análisis mínimo sobre las tareas a ejecutarse y delegarse.</li> </ul>	20%	Marginal.	<ul style="list-style-type: none"> <li>Demora del tiempo de ejecución de las tareas.</li> <li>Sobrecarga de responsabilidades a miembros del grupo de trabajo.</li> </ul>

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

Documento 12. Identificación de riesgos genéricos- Iteración 3.

**DOCUMENTO PARA IDENTIFICACIÓN DE RIESGOS****RIESGOS ESPECÍFICOS****NOMBRE DEL PROYECTO:** Titan**ITERACIÓN:** 3**FECHA:** 2011 – 01 – 31**VERSIÓN DEL DOCUMENTO:** 1.0

ID RIESGO	DESCRIPCIÓN	CAUSAS	% DE OCURRENCIA	NIVEL DE CRITICIDAD	CONSECUENCIAS
E8	Falta de acuerdo en las prioridades de las tareas, entre las áreas del departamento de sistemas.	<ul style="list-style-type: none"> <li>No lograr un acuerdo óptimo para priorizar actividades y requerimientos, debido a intereses y metas individuales de cada área.</li> </ul>	20%	Crítico.	<ul style="list-style-type: none"> <li>Priorizar de manera incorrecta las actividades, abarcando solamente la necesidades de un área.</li> </ul>
E9	Corregir manualmente errores de los usuarios.	<ul style="list-style-type: none"> <li>El cliente interno posee la idea de cada error cometido puede ser arreglado manualmente y de manera rápida.</li> <li>No tomar la responsabilidad necesaria al momento de ejecutar procesos.</li> </ul>	5%	Crítico.	<ul style="list-style-type: none"> <li>Acostumbrar a los usuarios a realizar las tareas de forma mecánica lo que puede incrementar la existencia de errores.</li> <li>Los usuarios no son capaces de encontrar una posible solución a los errores generados.</li> </ul>
E10	Planificación indebida de los proyectos y recursos a ser destinados.	<ul style="list-style-type: none"> <li>No contemplar a todos los involucrados del proyecto.</li> <li>No establecer los alcances necesarios para cumplir las expectativas del objetivo del proyecto.</li> </ul>	40%	Crítico.	<ul style="list-style-type: none"> <li>Vacíos en los escenarios que son analizados post desarrollo e incide en el tiempo de finalización del mismo.</li> <li>No cumplimiento de objetivos principales del proyecto.</li> </ul>

---

 FIRMA DE RESPONSABILIDAD

En la tarea 5, se generó un documento que agrupe tanto a riesgos genéricos como específicos, en el cual se analizó la priorización de atención que se dio a los riesgos, esta información se describe a continuación en el Documento 14:

<b>DOCUMENTO PARA PRIORIZACIÓN DE RIESGOS</b>				
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>				
<b>NOMBRE DEL PROYECTO: Titan</b>		<b>ITERACIÓN: 3</b>		
<b>FECHA: 2011 – 02 – 01</b>				
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>				
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>NIVEL DE PRIORIDAD</b>	<b>ORDEN DE ATENCIÓN</b>	<b>OBSERVACIONES</b>
G5	Asignación incorrecta de responsabilidades.	Medio.	1	
E8	Falta de acuerdo en las prioridades de las tareas, entre las áreas del departamento de sistemas.	Alta.	3	
E9	Corregir manualmente errores de los usuarios.	Alta.	4	
E10	Planificación indebida de los proyectos y recursos a ser destinados.	Alta.	2	

\_\_\_\_\_

FIRMA DE RESPONSABILIDAD

Documento 14. Priorización de riesgos - Iteración 3.

Se ejecutaron las tareas 6 y 7, concernientes a la etapa 3, planteamiento y gestión de soluciones; recopilando la información visualizada en el Documento 15, la que es necesaria para realizar la tarea 8.

<b>DOCUMENTO PARA GESTIÓN DE RIESGOS</b>					
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>					
<b>NOMBRE DEL PROYECTO: Titan</b>			<b>ITERACIÓN: 3</b>		
<b>FECHA: 2010 – 02 – 02</b>					
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>					
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>NIVEL DE GESTIÓN</b>	<b>SOLUCIÓN</b>	<b>TIEMPO ESTIMADO</b>	<b>RESPONSABLE</b>
G5	Asignación incorrecta de responsabilidades.	Mitigación de riesgos.	Distribuir equitativamente la carga de trabajo para cada miembro del grupo, tomando en cuenta sus habilidades y fortalezas.	2 horas.	Gerente / Jefe de área de proyectos.
E8	Falta de acuerdo en las prioridades de las tareas, entre las áreas del departamento de sistemas.	Mitigación de riesgos.	Analizar las necesidades que tienen en común las áreas y partir de los aspectos comunes para posteriormente abarcar las diferencias y llegar a un acuerdo sobre las necesidades prioritarias y que requieren una atención rápida.	3 horas.	Gerente / Jefe de área de proyectos - Vicepresidente / Gerente de área.
E9	Corregir manualmente errores de los usuarios.	Arreglar cada error.	Promover la conciencia en los usuarios finales que al realizar cualquier proceso se lo haga de forma lógica y no de forma mecánica y promover el uso de material de apoyo, sean estos manuales o experiencia de otras personas del área.	6 horas.	Grupo de desarrollo.
E10	Planificación indebida de los proyectos y recursos a ser destinados.	Mitigación de riesgos.	Hacer participar a todos los involucrados que intervienen en el proyecto (áreas, personas, recursos), para minimizar el impacto de falta de datos, documentación en la finalización del mismo.	3 horas.	Gerente / Jefe de área de proyectos.

\_\_\_\_\_  
FIRMA DE RESPONSABILIDAD

La tarea 9 que corresponde al control de la solución, se generó el Documento 16, que se describe a continuación:

<b>DOCUMENTO PARA CONTROL DE SOLUCIÓN DE RIESGOS</b>			
<b>RIESGOS GENÉRICOS Y ESPECÍFICOS</b>		<b>ITERACIÓN: 3</b>	
<b>NOMBRE DEL PROYECTO: Titan</b>			
<b>FECHA: 2010 – 02 – 04</b>			
<b>VERSIÓN DEL DOCUMENTO: 1.0</b>			
<b>ID RIESGO</b>	<b>DESCRIPCIÓN</b>	<b>ESTADO DE LA SOLUCIÓN</b>	<b>OBSERVACIONES</b>
G5	Asignación incorrecta de responsabilidades.	100%	
E8	Falta de acuerdo en las prioridades de las tareas, entre las áreas del departamento de sistemas.	70%	
E9	Corregir manualmente errores de los usuarios.	50%	
E10	Planificación indebida de los proyectos y recursos a ser destinados.	80%	

\_\_\_\_\_  
FIRMA DE RESPONSABILIDAD

Para finalizar el análisis de la tercera iteración se realiza la tarea 10 de la etapa 4 de la propuesta, llenando la información en el documento 6 que se generó anteriormente para el análisis de resultados de la gestión hecha a todos los riesgos generados en el proyecto.

#### **4.2.2. ANÁLISIS DE RESULTADOS.**

- El análisis y gestión de riesgos realizada para las tres iteraciones del desarrollo del sistema, que fueron tomadas como ejemplo, demuestran que la reducción de riesgos se ha logrado con la aplicación de la propuesta.
- A medida que el desarrollo avanzó, en cada iteración se evidenciaron menos riesgos, es así como en la primera iteración se identificaron 6 riesgos, en la segunda 5 riesgos, y en la última iteración en análisis se encontraron 4 riesgos.
- Los riesgos identificados para el análisis fueron tantos genéricos como específicos, para los que se brindaron soluciones que se acoplan a la realidad del ambiente del negocio, contando con la colaboración de las áreas involucradas.
- Para los riesgos identificados durante las tres iteraciones, se verificó que ninguno sobrepasa el porcentaje de ocurrencia del 50%, dado que es un rango manejable, se planifica que con la solución ese porcentaje reduzca de manera significativa.
- Aproximadamente el 90% de los riesgos identificados se relacionan en el factor humano, por lo al momento de plantear la solución, se intentó buscar que la misma satisfaga a las partes involucradas, y de esta manera lograr

un mejor ambiente que permita alcanzar que riesgos de ese tipo vayan reduciendo a medida que el proyecto avance.

- El nivel de gestión que se aplicó mayormente para los riesgos referidos fue Mitigación de riesgos con un 60%, seguido de Eliminación de causas principales con 27%, luego se encuentra Prevención y Arreglar cada error con un 13% cada uno, y finalmente Control de crisis con 7%; esto evidenció que se los riesgos presentados se solucionan de manera reactiva cuando suceden, intentando reducir su impacto en el proyecto.
- Los riesgos identificados en cada iteración se resolvieron en un 100% al final de cada sprint, logrando que estos no se repitan, por lo que el formato 7 de la propuesta no fue utilizado para el caso de estudio, y no hubo la necesidad de realizar un nuevo análisis sobre estos.
- La identificación oportuna de los riesgos alcanzada, ha permitido brindar la gestión adecuada con la aplicación de la propuesta, brindando así la solución más adecuada.
- La ejecución de la propuesta logró que el grupo de trabajo brinde un mayor nivel de importancia a la gestión de riesgos en el proyecto de desarrollo, lo que permitió que la propuesta sea implementada durante las siguientes iteraciones y además en el desarrollo de otros sistemas.

## **CAPITULO V.**

### **CONCLUSIONES Y RECOMENDACIONES.**

#### **5.1. CONCLUSIONES.**

- Las metodologías ágiles brindan muchos beneficios, tanto a las personas que las usan, como a los proyectos desarrollados con estas; logrando una mejor integración de todos los elementos, y así alcanzando mejor ejecución en cada etapa del proyecto, ya que brindan la importancia necesaria a cada componente del proyecto, permitiendo el desarrollo de todo como un conjunto y no como entidades independientes, que podrían no aportar de manera significativa al desarrollo del proyecto.

Estas metodologías se acoplan perfectamente a las características del mundo cambiante actual, tomando la poca previsibilidad de los proyectos como una ventaja, y los posibles cambios o modificaciones como oportunidades de mejora continua.

- Scrum es una de las metodologías más utilizadas actualmente para el desarrollo de diferentes proyectos de software, sin importar el tamaño y alcance del mismo, debido al grado de flexibilidad que brinda durante todo el proyecto. También es una de las metodologías que permite realizar una oportuna gestión de riesgos en el proyecto, ya que este proceso no es algo independiente a la metodología, sino es una actividad que se realiza durante cada etapa del proyecto ejecutado con esta metodología.
- El equipo multidisciplinario que posee Scrum como característica, permite que la toma de decisiones se realicen de forma ágil y de manera oportuna, logrando así que cada miembro del proyecto tenga la iniciativa correcta para aportarla durante el proyecto, además se fomenta la comunicación constante, ya que es difícil pensar que los proyectos de desarrollo puedan fluir de manera correcta sin

este intercambio de información continua; de manera que esto se convierte en algo habitual para los miembros del grupo de desarrollo.

- Una de las grandes ventajas que proporciona Scrum, es la entrega de productos ya funcionales en las primeras iteraciones, lo cual hace que el cliente visualice el compromiso y responsabilidad por parte del resto del grupo de desarrollo, lo que ayuda mucho para que la colaboración del cliente de forma óptima en el proyecto, ayudando al intercambio de información en todo sentido. Esto logra de mejor manera la inclusión del propietario del producto en el grupo de trabajo, que es lo fomenta Scrum; promoviendo la responsabilidad compartida para el éxito del proyecto. De esta manera por parte de las dos entes involucrados se plantea objetivos que van encaminados hacia la misma meta, sumando esfuerzos para lograrlos.
- Los riesgos en proyectos de desarrollo de software son un aspecto que no se puede evitar, pero la experiencia y la oportuna toma de decisiones de los miembros del grupo de trabajo pueden lograr que el impacto que producen estos sobre el proyecto sea mínimo o nulo. En ciertas situaciones estos podrían representar oportunidades para identificar las posibles debilidades en el proyecto, y así enfocar el esfuerzo del grupo de trabajo en reducir y eliminar estos inconvenientes.
- Las metodologías de software no son las que resolverán todos los problemas que se presenten durante los proyectos de desarrollo de software, ni son una guía de cómo resolver dichos inconvenientes; sino mas bien son una herramienta, que al usarla de manera adecuada para lograr su máximo beneficio, serán de mucha ayuda al momento de enfrentar un problema en cualquier tipo de proyecto.
- Durante la creación de la propuesta se ha logrado identificar la información necesaria que se encuentra relacionada con los riesgos presentados en los

proyectos, de esta manera a las personas que ejecuten la propuesta les resultará más fácil poder realizar la gestión de riesgos.

La ejecución de la propuesta ha logrado evidenciar que en el proyecto de desarrollo de software que se tomo como caso de estudio no se prestaba la debida importancia a la gestión de riesgos, tomando a esto como un proceso externo al proyecto, el cual solo es usado en casos extremos, y no precisamente como gestión, sino como una acción reactiva ante inconvenientes, donde no se analizan todos los posibles elementos involucrados en el problema, ya que solo se trata de enmendar los riesgos de forma que solo se solucione en ese momento, sin garantizar que no se vuelva a producir.

- Para el proyecto de desarrollo que se tomo como caso de estudio, fue de gran ayuda identificar los riesgos y poder diferenciarlos entre genéricos y específicos, ya que esto permitió que la gestión de riesgos se realice de forma organizada y oportuna.
- El número de riesgos identificados en cada una de las iteraciones no fue grande, ya que el tiempo entre cada iteración fue de dos semanas, este periodo ayudo para que el ambiente pueda ser mejor manejado al momento de identificar los riesgos y dar la solución respectiva.
- Para la empresa con la cual se trabajó en la implementación de la propuesta, la ejecución de la misma significó una opción muy viable para tomar en cuenta la gestión de riesgos dentro de los procesos que involucran el desarrollo de software en el negocio, además representa la oportunidad de adquirir una nueva base de conocimientos, que sirvan como experiencia tanto en próximas iteraciones del sistema Titan como en futuros proyectos.

La experiencia que significó la implementación de la propuesta para la empresa se redacta en el Anexo B.

## 5.2. RECOMENDACIONES.

- Las empresas al desarrollar proyectos de software deben tomar la metodología más adecuada, analizando las ventajas y desventajas que estas puedan brindar al entorno del proyecto, tomando muy en cuenta las características propias del negocio, ya que la metodología será la que debe adaptarse y acoplarse con estas características, y no que la empresa se adapte a la metodología seleccionada.
- Dado la mayor acogida que las metodologías ágiles tienen actualmente en el mercado, es muy aconsejable seleccionarlas para el desarrollo de proyectos de software, el alcance o tamaño del mismo no afectará en la eficiencia de la metodología, sino más bien dependerá de cómo los miembros del grupo de trabajo logran el acoplamiento de la metodología y el proyecto.
- Scrum es una de las metodologías ágiles que se acopla fácilmente con la gestión de riesgos, esta propone una constante revisión del proyecto, realizando un control sutil del mismo, permitiendo detectar y solucionar inconvenientes de manera oportuna; por lo que se recomienda a Scrum una opción muy viable al momento de seleccionar la metodología con la que se va a trabajar.
- Al momento de trabajar con Scrum se debe tomar en cuenta cuales son las responsabilidades que cada miembro debe cumplir en el proyecto, ya que en esta metodología no se maneja el concepto estricto de roles asignados a las personas, sino de compromisos que aceptaron asumir, y por ende cumplir; tomando como ejemplo las responsabilidades del Scrum Master deben ser asumidas por una persona que tenga la capacidad de poder mantener un ambiente cordial con el Propietario de Producto o cliente, para crear un ambiente de confianza y colaboración en el proyecto.  
Cada persona del equipo debe poseer la apertura necesaria para que exista comunicación constante, logrando el acoplamiento como grupo y a la vez la

formación de valores individuales como: seguridad, autocrítica, superación o autorrealización; pero no se debe confundir la autonomía que caracteriza al personal, con la opción de que exista poca capacidad de compartir información entre los miembros, sino más bien hay que entender que uno de los beneficios de esta metodología es que no solo fomenta el crecimiento individual sino que además busca que las habilidades de cada persona contribuya a los otros miembros del grupo, intercambiando información y conocimiento.

- La gestión de riesgos debe ser tomada como una actividad importante en los proyectos de desarrollo de software, por lo que es recomendable poseer lineamientos que vayan acordes con la metodología usada, para poder manejar los diferentes riesgos presentados y brindar la solución oportuna.
- Se recomienda usar la propuesta sugerida para reducción de riesgos basada en la metodología Scrum, de manera continua durante todas las iteraciones del desarrollo del sistema, de esta manera lograr que el grupo de trabajo se familiarice con la propuesta, y pueda ejecutarse con mayor fluidez y rapidez durante el proyecto, además hay que tomar en cuenta que esta puede ser aplicada sin importar el nivel del alcance del proyecto, ya que tanto la propuesta y la metodología pueden acoplarse a proyectos de grande o pequeño alcance.
- La propuesta sugerida puede ser usada en proyectos de cualquier tamaño, siempre y cuando sean desarrollados con la metodología Scrum, esta pueda ser acoplada a los requerimientos del proyecto, ya que brinda la misma adaptabilidad que la metodología en la fue basada su creación.

## REFERENCIAS BIBLIOGRAFICAS.

### REFERENCIAS EN INTERNET.

1. ¿Cuáles son las principales ventajas de Scrum como metodología?  
<http://es.w3support.net/index.php?db=so&id=208442>
2. Desarrollo ágil de software.  
[http://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)
3. El método Scrum.  
[http://www.mastersoft.com.ar/MsWeb/otros\\_archivos/NotaScrumPCUsers.pdf](http://www.mastersoft.com.ar/MsWeb/otros_archivos/NotaScrumPCUsers.pdf)
4. El rol de la arquitectura de Software en las Metodologías Ágiles.  
[http://www.epidataconsulting.com/tikiwiki/tiki-read\\_article.php?articleId=28](http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=28)
5. Flexibilidad con Scrum.  
[http://www.navegapolis.net/files/Flexibilidad\\_con\\_Scrum.pdf](http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf)
6. Gestión de proyectos con Scrum.  
<http://www.ingenierossoftware.com/equipos/scrum.php>
7. Gestión de Riesgos en la Administración de Proyectos.  
<http://arturoweb.wordpress.com/2008/02/29/gestion-de-riesgos-en-la-administracion-de-proyectos/>
8. Gestión de Riesgos en Proyectos Software.  
<http://alarcos.inf-cr.uclm.es/doc/pgsi/doc/teo/7/pgsi-t7.pdf>
9. Guía sobre Scrum.  
<http://www.scrum.org/storage/scrumguides/Gua%20sobre%20Scrum.pdf>
10. Identificación de Riesgos de Proyectos de Software en base a taxonomías.  
[http://www.itba.edu.ar/nuevo/archivos/secciones/http\\_\\_\\_www.centros.itba.edu.a..  
.niasi-Anteproyectodetesis.pdf](http://www.itba.edu.ar/nuevo/archivos/secciones/http___www.centros.itba.edu.a..niasi-Anteproyectodetesis.pdf)
11. La nueva metodología.  
<http://www.programacionextrema.org/articulos/newMethodology.es.html>
12. Los métodos ágiles.  
<http://tratandodeentenderlo.blogspot.com/2010/03/los-metodos-agiles.html>

13. Metodologías Ágiles.

<http://www.seccperu.org/files/Metodologias%20Agiles.pdf>

14. Metodologías Ágiles (introducción).

<http://www.gravitar.biz/index.php/bi/metodologias-agiles-intro/>

15. Metodologías Ágiles en el Desarrollo de Software.

<http://www.willydev.net/descargas/prev/TodoAgil.pdf>

16. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).

<http://www.cyta.com.ar/ta0502/v5n2a1.htm>

17. Metodologías Ágiles de Gestión de Proyectos (Scrum, DSDM, Extreme Programming – XP).

<http://www.marblestation.com/?p=661>

18. Principios del Manifiesto Ágil.

<http://www.ohagile.com/category/scrum/>

19. Proyectos Ágiles con Scrum.

<http://santimacnet.wordpress.com/2010/11/05/proyectos%C2%A0agiles-con-scrum/>

20. Que es Scrum.

<http://www.proyectosagiles.org/que-es-scrum>

21. Riesgos del Software.

<http://www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html>

22. Riesgos del Software.

[http://www.wikilearning.com/curso\\_gratis/gestion\\_de\\_riesgos\\_en\\_ingenieria\\_del\\_software-riesgos\\_del\\_software/3620-2](http://www.wikilearning.com/curso_gratis/gestion_de_riesgos_en_ingenieria_del_software-riesgos_del_software/3620-2)

23. Riesgos del Software.

<http://www.sitios.uach.cl/caminosfor/CristianSalazar/SIE/RS.html>

24. Scrum.

<http://www.dosideas.com/wiki/Scrum>

25. Scrum.

<http://www.slideshare.net/wilh3m/scrum-3760469>

26. Scrum.

<http://es.wikipedia.org/wiki/Scrum#Historia>

27. Scrum metodologías ágiles.

<http://vetepensandolo.files.wordpress.com/2008/06/05-scrum-metodologias-agiles.pdf>

## **ANEXOS.**

### **ANEXO A.**

Glosario de términos en inglés.

### **ANEXO B.**

Carta de la empresa – conclusiones sobre aplicación de la propuesta.

## ANEXO A.

### GLOSARIO DE TERMINOS EN INGLÉS.

- **Adaptive Software Development (ASD):** Desarrollo de software adaptable.
- **Agile Unified Process (AUP):** Proceso ágil unificado.
- **Burn up:** encender.
- **Burn down:** encender.
- **Crystal Clear:** Cristal Claro.
- **Essential Unified Process (EssUP):** Proceso esencial unificado.
- **Feature Driven Development (FDD):** Desarrollo impulsado por características
- **Lean Software Development (LSD):** Desarrollo de software impulsado.
- **Open Unified Process (OpenUP):** Proceso abierto unificado.
- **Product Backlog:** listado de requerimientos del producto.
- **Sprint:** iteración.
- **Sprint Backlog:** lista de requerimientos de la iteración.

## **ANEXO B**

Quito 12 de Marzo del 2011

Proyecto Titán

El proyecto Titán y su implementación se inició a mediados del año 2007 con tres equipos de trabajo. EL equipo 1 que tiene que ver con las actividades de recolección de información, realización de flujos de proceso y optimización, el equipo 2 que tenía a cargo la parte de desarrollo y el equipo 3 la parte de pruebas. En un inicio de acuerdo a los problemas presentados en la matriz de esta tesis, tubo bastantes dificultades por la falta de comunicación, no exigencia de parámetros o estándares mínimos requeridos al proveedor, planificación incorrecta de las fases del proyecto, asignación de recursos poco acordes a la realidad del proyecto.

A partir de los errores encontrados que hacían extender los tiempos de desarrollo, se requirió de la conformación de un comité de proyecto que debía analizar y detectar las causas de los problemas para de esta manera establecer soluciones definitivas y no parches, con la ayuda de la propuesta para reducir riesgos en proyectos de software, lo que también permitió cumplir los tiempos establecidos para el desarrollo de todos los procesos que el usuario necesite para realizar de mejor manera sus actividades.

De igual forma la propuesta permitió reducir los tiempos de ciclo del proyecto, con el establecimiento de estándares de cumplimiento, al mismo tiempo de reducir los costos de implementación al solicitar en cada requerimiento un análisis costo beneficio de poner en producción X desarrollo y sobretodo que las necesidades generen valor a la compañía. Estos beneficios adquiridos por el proyecto del desarrollo del sistema Titan, han ayudado a reducir los riesgos en el mismo.

Actualmente las plataformas en las que se respaldan todos los sistemas de gestión se encuentran trabajando eficientemente, con tiempos de respuesta acorde a las necesidades del usuario y del cliente y se va a ingresar a una etapa de optimización de todos los procesos en la cadena de valor de la compañía , razón por la cual se implementará la propuesta de reducción de riesgos para otros proyectos de software, adicionales al sistema Titan, para así cumplir los objetivos corporativos y del área, de acuerdo a la planificación estratégica realizada para inicios de este año cuyo objetivo es mantener a los clientes satisfechos con los servicios que presta el grupo, generando valores añadidos a los productos que oferta con alta capacidad de respuesta y compromiso en la gestión.

Lenin Villareal