REPUBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD CENTRO OCCIDENTAL
“LISANDRO ALVARADO”

MODELO DE MEJORA CONTINUA PARA LA GESTIÓN DE REQUISITOS DE SOFTWARE USANDO MÉTODOS ÁGILES.

OSCAR EDUARDO IRIBARREN PAEZ

Barquisimeto, 2012

UNIVERSIDAD CENTROOCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGIA
POSTGRADO EN CIENCIAS DE LA COMPUTACION

MODELO DE MEJORA CONTINUA PARA LA GESTIÓN DE REQUISITOS DE SOFTWARE USANDO MÉTODOS ÁGILES

Trabajo presentado para optar al grado
Magister Scientiarum

Por: OSCAR EDUARDO IRIBARREN PAEZ

Barquisimeto, 2012

DEDICATORIA

A Dios todo poderoso porque sin el nada de esto hubiese sido posible.

A mis padres, esposa e hijo por ser mi mayor fuente de inspiración.

AGRADECIMIENTOS

A la santísima trinidad por haberme dado toda la fuerza, sabiduría y constancia para poder terminar este proyecto.

A mis padres por haberme inculcado desde pequeño la importancia de los estudios y por su apoyo incondicional en los momentos difíciles.

A mi esposa María Auxiliadora por llenarse de paciencia en estos 4 años de estudio donde parte del tiempo era dedicado a este trabajo y por su apoyo para la culminación del mismo.

A mi hijo Jesús Antonio por haber nacido este año y traerme la alegría y motivación que necesitaba en este último año de estudio.

A mi tutor Ramón Valera por la orientación, guía y el aporte de ideas para la realización de este trabajo.

A los profesores Jorge Pérez, Edison Sira y Ana Mercedes. Díaz por sus revisiones, ideas y aportes a este trabajo.

A todos los profesores de postgrado de la universidad por impartir sus conocimientos a lo largo de este estudio.

A todos aquellos compañeros de postgrado que de una u otra forma aportaron su granito de arena para este aprendizaje.

A todos aquellos amigos y familiares que estuvieron pendientes de este trabajo y dieron su energía positiva y aportes para la culminación de esta investigación.

INDICE GENERAL

INDICE DE TABLAS

pp

INDICE DE FIGURAS

pág.

UNIVERSIDAD CENTROOCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGIA
POSTGRADO EN CIENCIAS DE LA COMPUTACION

MODELO DE MEJORA CONTINUA PARA LA GESTIÓN DE REQUISITOS USANDO MÉTODOS AGILES.

Autor : Oscar Eduardo Iribarren Páez
Tutor: Ramón Valera
Fecha: Septiembre 2012

RESUMEN

En las empresas actuales dedicadas al desarrollo de software es de vital importancia lograr la adecuación a los cambios de manera rápida, dependiendo de esta capacidad se podrán satisfacer o no las necesidades de los clientes, ellas necesitan modelos que permitan gestionar requisitos de una manera ágil para poder adaptarse a las exigencias de la vida moderna. El presente trabajo de investigación tiene como objetivo proponer un modelo de mejora continua para la gestión de requisitos soportado en la ingeniería de métodos y usando métodos ágiles, para ello se estudiara la metodología de desarrollo ágil Scrum para conocer la manera en que se lleva a cabo la gestión de requisitos, de forma que pueda generarse un cuadro que muestre las virtudes de las actividades de ingeniería de requisitos que se desarrollan en esta metodología, con esto se espera tomar las características más relevantes para la elaboración del modelo propuesto, después de tener definido el modelo este se modelara en una herramienta de modelaje de procesos de negocio (BPMS) y luego se diseñaran indicadores de desempeño del modelo obtenido que puedan servir de base para impulsar un proceso de mejora continua.

Palabras Claves: Ágil, Gestión de Requisitos, Ingeniería de métodos, Mejora continua, Manejo de Procesos de Negocio.

INTRODUCCION

El proceso de desarrollo de software es un conjunto de actividades que se complementan para obtener un producto, entre las actividades fundamentales de este proceso podemos mencionar la especificación, donde se define el software a producir y las restricciones para su operación; el desarrollo de software, donde se diseña o programa; la validación donde el software se válida para asegurar que es lo que el cliente desea y la evolución que es cuando el software se modifica para adaptarlo a los cambios que el cliente o el mercado requieren.

Estos procesos de desarrollo han experimentado una serie de cambios en la manera de tratar las actividades a lo largo del proceso, a partir del final del año 1990 y durante la década actual los modelos son cada vez más adaptativos y menos predictivos, estos modelos basan su filosofía en escribir código rápidamente y realizar revisiones periódicas con los clientes en vez de tratar de anticipar y documentar todos los requisitos en el comienzo del proceso de desarrollo de software, Sommerville (2005).

En este mismo orden de ideas, en la gestión de requisitos usada en los métodos de desarrollo ágil se puede observar dos aspectos diferentes en comparación con los métodos tradicionales; el primero que se observa es que la máxima prioridad es satisfacer al cliente a través de la entrega temprana y continúa de funcionalidades de valor para el cliente y segundo que son bienvenidos los cambios según las necesidades detectadas inclusive al finalizar el proceso de desarrollo. El impacto en la industria del software de estos nuevos métodos ha sido significativo, con la agilidad se toma un enfoque mucho más flexible para la gestión de requisitos que es mucho más temporal, interactiva y justo a tiempo.

En este sentido, el propósito fundamental de esta investigación está orientado a la elaboración de un modelo de mejora continua para la gestión de requisitos de software usando metodologías ágiles que sirva a las empresas desarrolladoras de software para mejorar su productividad y eficiencia; para lograrlo se pretende estudiar las mejores prácticas de la metodología Scrum en lo que corresponde a la gestión de requisitos, de forma que se pueda diseñar un modelo con las mejores prácticas; posteriormente se usara una herramienta BPMS para modelar el modelo obtenido y finalmente se diseñaran indicadores de desempeño para el modelo propuesto con el fin de impulsar un proceso de mejora continua que permita mejorar la automatización de los procesos, evaluación de los procesos y la toma de decisiones .

Con el fin de dar cumplimiento de esta investigación se estructura la misma de la manera siguiente:

En el Capítulo I, El Problema, se plantea la problemática existente en las empresas desarrolladoras de software en cuanto a la gestión de requisitos, así como también los objetivos que se persiguen con la investigación y la justificación e importancia de la misma.

En el Capítulo II Marco Teórico, se presentan los antecedentes de la investigación que sirven como marco referencial para el estudio y los basamentos teóricos sobre los cuales se fundamenta el mismo.

En el Capítulo III Marco Metodológico, se expone la metodología aplicada, contentiva de la modalidad y diseño del estudio y las herramientas usadas en el proceso de recolección de datos.

CAPITULO I

EL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

En los años 80 y principios de los 90, existía una opinión general de que la mejor forma de obtener un mejor software era a través de una planificación cuidadosa del proyecto. Esta opinión provenía, fundamentalmente, de la comunidad de Ingenieros de Software implicada en el desarrollo de grandes sistemas de Software que normalmente se componían de un gran número de programas individuales, Sommerville (2005).

En estos años, existía el paradigma entre los expertos de la Ingeniería del software de que la mejor forma de obtener un mejor software era a través de una planificación exhaustiva de los proyectos usando procesos de desarrollo de software controlados y rigurosos. Este paradigma significa una importante sobrecarga de trabajo en cuanto a la planificación, diseño y documentación del sistema; este gran esfuerzo se justifica cuando el software es un sistema crítico y cuando existen muchas personas involucradas en el mantenimiento del software durante su ciclo de vida; para el caso de equipos de pocas personas esta forma de trabajar es sumamente pesada.

Es por ello que en los años 90 varios desarrolladores de software proponen nuevos métodos ágiles que permitieron a los equipos de desarrollo centrarse en el software en vez de su diseño y documentación, Sommerville (2005). Los métodos ágiles dependen de un enfoque iterativo para la especificación, desarrollo y entrega del software,

Y principalmente fueron diseñados para apoyar el desarrollo de aplicaciones de negocio donde los requisitos del sistema normalmente cambiaban rápidamente durante el proceso de desarrollo, Sommerville (2005).

En este contexto las metodologías ágiles aparecen como una opción atractiva en contraposición con las metodologías convencionales que actúan basadas en principios de estabilidad y control del contexto, las metodologías ágiles no se centran en habilidades de predicción, ni pretenden tener un sistema perfectamente definido como paso a su construcción sino que perciben cada respuesta al cambio como una oportunidad para mejorar el sistema e incrementar la satisfacción del cliente, considerando la gestión de cambios como un aspecto inherente al propio proceso de desarrollo de software y, permitiendo de este modo, una mejor adaptación en entornos turbulentos, Rodríguez, P. (2008).

Los métodos ágiles universalmente dependen de un enfoque iterativo para la especificación, desarrollo y entrega del software, y principalmente fueron diseñados para apoyar el desarrollo de aplicaciones de negocio donde los requisitos del sistema normalmente cambiaban rápidamente durante el proceso de desarrollo. Están pensados para entregar software funcional de forma rápida a los clientes, quienes pueden entonces proponer que se incluyan en iteraciones posteriores del sistema nuevos requisitos o cambios en los mismos, Sommerville (2005).

El desarrollo ágil de software define nuevas metodologías para afrontar su desarrollo de una forma más eficiente, y por tanto menos costosa. Estos métodos, llamados inicialmente ligeros, por contraposición a los tradicionales métodos pesados, asumen el cambio como algo inevitable, para lo cual se hace necesaria una nueva forma de abordar el desarrollo de software que

facilite el acomodo a los nuevos Requisitos a medida que éstos surjan, en vez de pretender analizar inicialmente el dominio a modelar de forma tan exhaustiva que ya no se produzca luego ningún cambio o estos sean mínimos, Pelayo (2007).

De acuerdo a Ros (2009). La Ingeniería de Requisitos influye decisivamente en el resto de los procesos de desarrollo y mantenimiento de un sistema. No es un proceso que se realiza solo en las primeras etapas del desarrollo de un proyecto y produce un conjunto de documentos que quedan "congelados". Dado que los Requisitos son de naturaleza cambiante de forma inevitable se encuentran sujetos a cambios constantes durante todo el desarrollo; es por tanto necesario gestionar adecuadamente los cambios en los Requisitos a lo largo del desarrollo, la explotación y el mantenimiento del software, a través de sus relaciones de traza con el resto de los productos de desarrollo y con los compromisos que se hayan establecido.

Es por ello que la ingeniería de Requisitos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir, Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta el comportamiento del sistema. Como disciplina, establece el proceso de definición de requisitos en una sucesión de actividades mediante las cuales lo que debe hacer, se elicita, se modela y analiza (Choque, 2001) en Merchán L (2008).

Por su parte Berrocal (2009) mencionó que en los últimos años todas las empresas están realizando grandes esfuerzos para ser cada vez más competitivas. Las compañías están buscando nuevas formas que les permitan mantenerse como los líderes del mercado actuando ágilmente ante los posibles cambios del mismo.

Persiguiendo este objetivo, las empresas han ido adoptando conceptos como BPM (Manejo de Procesos de Negocio) que las guían hacia un mayor control y gestión de sus procesos de negocio para obtener dicha agilidad. Los procesos de negocio han ido adquiriendo a lo largo de los años un mayor protagonismo en las empresas desarrolladoras de software, las organizaciones están constantemente evaluando y mejorando sus procesos de negocio para que proporcionen la máxima rentabilidad al negocio (Berrocal y otros, 2010)

(Berrocal y otros, 2010) en Sivira (2011) afirma que Una de las ventajas de usar BPM, es que es posible la automatización de muchas de las actividades asociadas a estos procesos de una forma sencilla y rápida, usando los BPM System (BPMS), disminuyendo el esfuerzo y aumentando la productividad de los gestores. Estos sistemas permiten realizar constante seguimiento a la ejecución de los procesos facilitando la recolección de indicadores que contribuyen a la mejora continua de estos procesos de negocio, traduciéndose en la ganancia de madurez, Sivira (2011).

Garimella (2008), menciona que BPM puede automatizar la ejecución de muchas tareas de procesos que pueden haber sido controlados anteriormente de forma manual; además permite detectar el cambio cuando se produce, interpretar el impacto de ese cambio y desarrollar una comprensión compartida sobre cómo debe responder la organización.

Zalghadar (2004) en Sivira (2011) hace referencia a que muchas organizaciones se enfrentan al problema de encontrar y definir indicadores en función de ser capaz de evaluar la calidad de sus procesos de negocio, por lo que es necesario contar con una estrategia que permita revisar de manera periódica la correcta ejecución de dichos procesos. Por esta razón, se hace relevante complementar la definición de los procesos con el uso de una herramienta que permita evaluar y monitorear la ejecución de los procesos y que a su vez genere los indicadores que apoyen la toma de decisiones en cuanto a las acciones a seguir frente a una detección de fallos o detección de mejoras, que atendiendo oportunamente podrían generar aún mejores resultados para la organización.

Silva (2005) en Sivira (2011), indica que la herramienta BPMS le permite a las organizaciones mapear, integrar, liberar, medir, monitorizar, controlar, analizar y optimizar procesos de negocio claves en las organizaciones y la integración de estos con la cadena productiva para finalmente generar valor a los usuarios finales y destinados al logro de objetivos estratégicos.

Analizando BPM y el desarrollo ágil de Software observamos que ambos tienen similitudes en su naturaleza, por ser ambos facilitadores de la respuesta al cambio, BPM desde una perspectiva empresarial y desarrollo ágil de software desde el punto de vista productivo.

Comúnmente los proyectos de software suelen tener problemas con la interacción entre su aplicación y la lógica del negocio donde se está desarrollando. Esta problemática se puede presentar porque se manejan múltiples fuentes de información, esto debido a la forma como está constituida la información o el poco conocimiento que tienen los desarrolladores de software del negocio de la lógica del negocio de la

empresa. Berrocal, J. (2009) menciona que existen dificultades para lograr y mantener una perfecta alineación entre los procesos del negocio y los sistemas IT, provocando una disminución de la eficacia de los sistemas IT y por lo tanto del negocio y una menor agilidad ante los cambios requeridos. Esta dificultad para conseguir la alineación viene producida por:

- Dificultades de comunicación entre stakeholders e Ingenieros de Requisitos debido al uso de distintos lenguajes, puntos de vista, etc.

- Stakeholders que solo tienen una visión parcial del negocio, lo que provoca que los ingenieros de Requisitos no tengan una clara visión del negocio y dificulta la elicitación de los Requisitos.

- Ingenieros de Requisitos que se centran en especificar los Requisitos a través de casos de uso, historias de usuario, etc, describiendo las funcionalidades pero perdiendo información sobre las relaciones entre Requisitos o de éstos con el entorno.

El Standish Group (2009) en González (2011), afirma que 32 % de los proyectos de software a nivel mundial son exitosos y de los que fracasan el 21,8 % corresponde a dificultades con la obtención de requisitos y al manejo apropiado a los cambios de estos durante el proceso de desarrollo; lo que significa que algunos de los problemas que llevaron al anuncio de una crisis del software en los años 60 del siglo pasado aún permanecen en el presente.

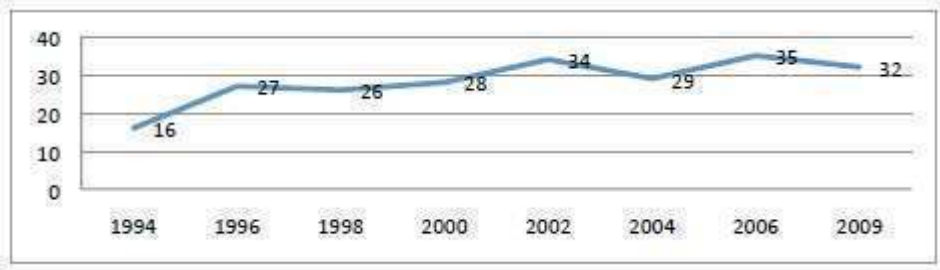


Figura 1. Proyectos exitosos según investigación de standish group.
Fuente: González (2011)

Este trabajo de investigación tiene como finalidad crear un modelo para mejorar algunos de los problemas que se mencionan en el estudio del Standish Group, entre los cuales está, el manejo apropiado a los cambios de estos durante el proceso de desarrollo.

Por lo anteriormente expuesto, se realizó una revisión bibliográfica de trabajos de grado, tesis y artículos relacionados con el tema de estudio entre los cuales se mencionan a continuación los más relevantes:

Merchán y otros (2008) en su artículo denominado “Implementación de modelos livianos de gestión de requisitos y diseño de un modelo liviano de apoyo al área de aseguramiento de la calidad del proceso de desarrollo de software para pequeñas empresas del Valle del Cauca”, mencionan que como industria, el software requiere de productos y servicios de alta calidad, lo cual se logra mediante la aplicación de modelos y metodologías de calidad reconocidos internacionalmente. Las empresas emergentes no logran aplicar estas metodologías pues su gran obstáculo se observa en los altos costos de implementación, el recurso humano requerido y los estándares exigidos que restringen la creatividad, parte importante de su capital.

Así mismo Merchán y otros (2008) afirma que un factor estratégico para las empresas desarrolladoras de software debe ser la aplicación de modelos

de mejoramiento de procesos, que una vez adoptados en los proyectos de software puedan disminuir los costos de producción y la inversión de recursos en el mantenimiento de los respectivos productos y servicios. Los modelos y metodologías actuales son extranjeros y ajenos a las condiciones y características propias, con servicios de capacitación muy formales y servicios de consultoría costosos y difíciles de aplicar en organizaciones pequeñas.

Por otro lado Sivira, B (2011), en su trabajo de grado menciona en su caso de estudio que entre los problemas detectados se encuentra la dificultad para identificar las fallas o mejoras en los procesos, las cuales no siempre son atendidas oportunamente, lo que ocasiona retardos y pocos insumos a la hora de rediseñar procesos como soporte a la mejora continua por lo que plantea una integración entre CMMI, BPM y BPMS para buscar dar solución a este problema.

En este mismo orden de ideas González O. (2011), en su artículo un acercamiento a la trazabilidad en el desarrollo ágil de software afirma que la trazabilidad de requisitos es una actividad importante en el desarrollo de software, tanto para las posteriores etapas de mantenimiento como para analizar el impacto de cambios de requisitos por parte del cliente; además de que constituye un elemento importante en el proceso de mejora continua de la organización. Así mismo González O. (2011) señala que mantener la trazabilidad de requisitos genera un costo adicional al proceso pues se carece de herramientas eficaces que automaticen las actividades de registro y seguimiento de trazas.

A raíz de estas y otras investigaciones se pudo precisar la necesidad de un modelo, para la gestión de requisitos que mitigue las siguientes situaciones:

- Cuando ya se está en la etapa de la programación del software se encuentra que a lo largo de esta etapa los Requisitos cambian constantemente y si no se está preparado para estos cambios los desarrollos se alargan mucho en los tiempos de entrega debido a estos cambios, esto puede ocasionar que los proyectos sean abandonados por no terminarse a tiempo o por superar los presupuestos asignados al mismo.

- No se puede realizar un seguimiento o trazabilidad de los Requisitos al momento de que estos son cambiados o modificados, ya que no se lleva a cabo la administración de los cambios en los requisitos y no se controla la relación con los demás requisitos y como pueden verse éstos afectados por el cambio.

- Los procesos que intervienen en los Requisitos del software a realizar no pueden ser simulados antes de ser desarrollados.

Es por lo planteado anteriormente que el autor considera pertinente realizar un análisis acerca de la manera como se maneja la etapa de requisitos de usuario en los métodos ágiles, para obtener de ellos las mejores prácticas que sirvan para diseñar un modelo de mejora continua para la gestión de requisitos usando metodologías ágiles, igualmente diseñar indicadores de desempeño que sirvan como base para la implementación de un sistema de mejora continua que permita evaluar los procesos y ayudar en la toma de decisiones dentro de las empresas de desarrollo de software.

Objetivos de la Investigación

Objetivo General

Diseñar un modelo de mejora continua para la gestión de requisitos de software usando métodos ágiles.

Objetivos Específicos

1. Realizar un estudio que muestre las prácticas más relevantes y actividades de gestión de requisitos de la metodología ágil Scrum.

2. Diseñar el modelo de mejora continua de gestión de requisitos usando ingeniería de métodos y basado en las mejores prácticas ágiles de la metodología Scrum descubiertas en el objetivo anterior.

3. Descubrir la arquitectura tecnológica necesaria para modelar los procesos de gestión de requisitos usando el lenguaje de modelado BPMN.

4. Diseñar los indicadores de desempeño para el modelo propuesto.

Justificación e importancia.

El presente trabajo nace de la necesidad de lograr que las empresas de desarrollo de Software actuales tengan la posibilidad de impulsar la mejora continua y satisfacer las peticiones de sus clientes aplicando modelos basados en metodologías ágiles y manejo de procesos de negocio para ser usados en la gestión de Requisitos de sus proyectos de desarrollo de software.

El uso de modelos para gestión de Requisitos de manera ágil junto con el manejo de los procesos del negocios son de gran importancia en las organizaciones para ayudar en la realización de proyectos de Software acordes con las necesidades de los usuarios y minimizar los riesgos de futuros problemas y fracaso de los proyectos. De allí surgirán indicadores que permitirán a la gerencia tomar decisiones orientadas a la mejora continua.

El diseño de un modelo de gestión de Requisitos de Software ágil basado en gestión de procesos de negocio para las empresas desarrolladoras de Software, cuyo diseño se desarrolle bajo una metodología, con el fin de lograr un producto de calidad, su aporte sería de gran importancia para dichas empresas, ya que facilitaría y serviría de guía para la ciclo de gestión de Requisitos de los proyectos de Software.

Alcances

El alcance de la presente investigación es realizar un modelo de mejora continua para la gestión de Requisitos de software de manera ágil aplicando prácticas de Scrum, basado en la ingeniería de métodos y modelado bajo la notación BPMN, que permita contribuir a la mejora continua identificando fallas y a partir de estos resultados poder mejorar los procesos seleccionados y que pueda ser aplicado en empresas donde se desarrolle software para mejorar la eficiencia y calidad en su manera de gestionar los Requisitos y finalmente sus productos puedan satisfacer las necesidades de sus clientes.

El autor del presente trabajo estudiara como se lleva a cabo la ingeniería de Requisitos en la metodología ágil Scrum para así obtener un cuadro con la mejores prácticas de la misma, también se diseñara un modelo de gestión de Requisitos apoyado en la teoría de ingeniería de métodos y aplicando prácticas de Scrum. Adicionalmente se plantea modelar el modelo obtenido bajo un enfoque BPM en una herramienta BPMS para comprender los procesos del modelo y por último diseñar los indicadores de desempeño del modelo propuesto que faciliten la medición del mismo y colaborar con la mejora continua de sus procesos.

Es importante destacar que esta investigación parte del hecho de que la captura de los requisitos ya se ha realizado, enfocándose solamente en la gestión de requisitos.

CAPITULO II

MARCO TEORICO

ANTECEDENTES

Dentro de las investigaciones realizadas que se refieren a Metodologías de gestión de requisitos de Software, Metodologías ágiles, Ingeniería de requisitos y al manejo de procesos de negocio (BPM), existen diversos estudios orientados a estos temas. Al respecto se han considerado un conjunto de investigaciones que constituyen antecedentes para el presente trabajo de investigación.

Para el entendimiento de los antecedentes y de su relación con el presente trabajo se toma como guía el cuadro referencial en la tabla 1 propuesto por Rolland et al (1998) en Sánchez I (2011) que establece cuatro vistas de un escenario o situación cada una de ellas permite la captura de los aspectos relevantes de cada escenario. En el caso de este trabajo se tomaron las siguientes vistas, preguntas y criterios:

Tabla 1. Cuadro de referencia para los antecedentes.

Vista	Pregunta	Criterio
Qué?	Cual fue el motivo de la investigación?	Objeto de estudio
Cómo?	Que procedimiento se llevo a cabo?	Metodología
	Que herramientas se usaron?	Tecnología
Cuando?	A quienes está dirigida la investigación?	Caso de estudios
Para qué?	Cuáles fueron las conclusiones del estudio?	Resultados
	En que se sustenta la presente investigación?	Aporte

Fuente: Rolland et al (1998) en Sánchez I (2011) .

Las investigaciones relacionadas con el presente trabajo se muestran en orden cronológico comenzando con la más antigua.

Merchán L y otros (2008). **Definición de una metodología ágil de ingeniería de requisitos para empresas emergentes de desarrollo de software del sur-occidente colombiano.** Este es un artículo del grupo de investigación para el desarrollo de la ingeniería de software (LIDIS), el motivo de esta investigación fue proponer un modelo liviano de mejoramiento de los procesos de desarrollo de software partiendo de la caracterización de empresas emergentes.

El procedimiento que se llevo a cabo fue primeramente una revisión de la literatura pertinente seguida de un diagnóstico para las empresas emergentes, basándose en la muestra de población y el instrumento de recolección de datos, luego se presenta un análisis de los datos para determinar los factores del problema en el procesos de ingeniería de requisitos.

La investigación está dirigida a las pequeñas empresas de desarrollo de software del Valle del Cauca en Colombia.

Las conclusiones del estudio son las siguientes; se observo la necesidad de las empresas emergentes de desarrollo de software del sur occidente colombiano de usar metodologías como las que se proponen en este artículo. Se afirma que la gran mayoría de empresas asegura que emplean metodologías de desarrollo pero realmente no realizan esas actividades de la forma correcta, o en otros casos no las realizan, lo que ocasiona problemáticas en cuanto a criterios para la aceptación de proveedores de requisitos, criterios para la aceptación de requisitos y ausencia de administración de trazabilidad y de los cambios en los requisitos. Finalmente se definió y documento una metodología ágil de ingeniería de requisitos para las empresas emergentes la cual comprende tres fases elicitación, especificación y gestión de requisitos.

Esta investigación fundamenta el presente estudio, ya que desarrolla una metodología ágil de ingeniería de requisitos y una de las fases que comprende es la gestión de requisitos, la cual sirve de guía para la elaboración del modelo que se quiere realizar.

Rodríguez, P. (2008). **Estudio de la aplicación de metodologías ágiles para la evolución de productos de software.** Tesis de Máster en tecnologías de la información, presentada en la facultad de informática de la Universidad Politécnica de Madrid. La finalidad de esta investigación es

estudiar la evolución de un producto de software utilizando las directrices marcadas por metodologías ágiles en concreto por la metodología SCRUM.

Entre una de sus conclusiones el autor señala que detecto que evolucionar un producto usando una metodología ágil repercute positivamente en la satisfacción del cliente debido a que se consigue una continua aportación de valor a su negocio, al permitir de forma progresiva ir disfrutando de la funcionalidad del producto.

Este trabajo realiza un estudio de la metodología ágil Scrum por lo que sirve de aporte teórico respecto a esta metodología.

Fernández, Y. (2009). **Estudio sobre la correspondencia entre prácticas CMMI y prácticas ágiles y su aplicación en pymes.** Tesis de Máster en tecnologías de la información, presentada en la Universidad Politécnica de Madrid. Esta tesis tiene como objetivo principal realizar un estudio sobre las correspondencias teóricas entre las prácticas ágiles y CMMI, proporcionando datos empíricos que confirmen correspondencias teóricas entre las prácticas ágiles y las metas específicas CMMI, que permita a pequeñas y medianas empresas implantar procesos de desarrollo de software conformes a CMMI a través de métodos más ligeros. Fernández, Y. (2009) menciona que la producción ágil de software representa un cambio de paradigma en la industria, demostrando su efectividad en entornos turbulentos y en proyectos con requisitos muy cambiantes.

Esta investigación está dirigida a las pequeñas y medianas empresas que no pueden asumir el costo de una implantación CMMI, el caso de empresas que han adoptado el modelo agile y necesitan certificarse, empresas que tienen implantado cierto nivel del modelo de procesos CMMI y quieren aplicar prácticas ágiles y viceversa.

Las conclusiones de este estudio es que el caso de estudio que consiste en la evaluación de un proyecto ágil, ha proporcionado evidencias de que las áreas de proceso planificación de proyecto, monitorización y control de proyecto y gestión de requisitos (Nivel 2 de CMMI) son, en gran parte soportadas a través de prácticas ágiles.

Una de las tres áreas en las que Fernández, Y. (2009) se centra es en la gestión de requisitos por lo que sirve de guía para la presente tesis ya que se busca un modelo que permita la gestión de requisitos, además de que Fernández, Y (2009) aporta a la presente investigación análisis de que manera Scrum soporta la gestión de requisitos y maneja la mejora continua de sus procesos.

Berrocal, J y otros (2009). **Patrones para la extracción de Casos de Uso a partir de Procesos de Negocio.** Artículo de la Universidad de Extremadura, Actas de los talleres de las jornadas de ingeniería de software y bases de datos, Vol. 3, No. 3, 2009. En este artículo se proponen una serie de patrones que permiten identificar los casos de uso del sistema IT a partir de los procesos de negocio de la organización. Con ello se contribuye a mejorar la alineación entre los procesos de negocio y los sistemas IT y la trazabilidad entre los procesos de negocio y los sistemas, mejorando la evolución entre ellos. Berrocal, J y otros (2009) menciona como pudo observar que a través de las guías y patrones propuestos en su trabajo se facilita la identificación de los requisitos del sistema a desarrollar. Por lo tanto, los requisitos deducidos estarán totalmente alineados con los procesos de negocio de la organización y, consecuentemente, los sistemas desarrollados lo estarán tanto con los requisitos como con los procesos de negocio, contribuyendo a mejorar la alineación negocio-IT. Además estos patrones ayudan a mantener la trazabilidad entre los procesos de negocio y

los requisitos deducidos, facilitando que la evolución del negocio se refleje en el sistema.

Berrocal, J y otros (2009). Concluyo que en los últimos años han ido adquiriendo cada vez más importancia los conceptos BPM y que las empresas se han dado cuenta que los sistemas que sustentan sus negocios deben cubrir sus necesidades los mas adecuadamente posible, y que cualquier evolución del negocio debe ser correctamente reflejada en estos sistemas, necesitando, por tanto, metodologías que permitan desarrollar sistemas que estén alineados con el negocio y que sean capaces de mantener una correcta trazabilidad, para que cualquier evolución del negocio sea transmitida a los sistemas.

Este trabajo sirve de aporte teórico a la presente investigación ya que analiza la necesidad de metodologías alineadas con las necesidades del negocio y conceptos importantes de BPM.

Sánchez, I (2011). **Extensión del método Gray Watch basado en el estándar de calidad ISO/IEC 25010.** Proyecto de grado para optar al título de Magister en Ciencias de la computación en La Universidad Centroccidental Lisandro Alvarado. Barquisimeto Edo. Lara – Venezuela. La presente investigación tuvo como finalidad diseñar la extensión del método GRAY WATCH en su más reciente versión propuesta por Montilva (2008), específicamente en los procesos técnicos de Análisis y Diseño, a su vez los productos generados en los Procesos de Análisis y Diseño se acoplaron en el Proceso de Implementación.

La naturaleza del estudio se enmarcó en la modalidad de proyectos especiales y se utilizo la nomenclatura SPEM para la realización de los

diagramas de los procesos y como caso de estudio se presento el estudio del programa de investigación de la UPEL IPB.

Como conclusiones se tiene que La propuesta presentada en este estudio incorporo al método GRAY WATCH la aplicación del estándar ISO/IEC 25010, lo que permitió profundizar acerca del modelo de productos más adecuado que describe los artefactos generados en el proceso de calidad al aplicar esta norma. De igual forma, se pudo conocer los pasos a seguir para obtener los requisitos de calidad, modelos de calidad y el diseño arquitectónico. Para ello, se modificaron y extendieron subprocesos del proceso de Análisis y Diseño de la aplicación del método GRAY WATCH, los procesos y los productos generados se adaptaron al resto de los procesos de la Cadena del método GRAY WATCH.

Este trabajo sirve de aporte teórico a esta investigación ya que realiza un estudio de la ingeniería de métodos, por lo que sirve de guía en la presente investigación para la construcción del modelo de mejora continua usando ingeniería de métodos.

Sivira, B. (2011). **Diseño de un plan de mejora continua en un proceso de desarrollo de software basado en RUP y usando prácticas BPM y CMMI.** Proyecto de grado para optar al título de Magister en Ciencias de la computación en La Universidad Centroccidental Lisandro Alvarado. Barquisimeto Edo. Lara – Venezuela. En este trabajo se define un plan de mejora continua para El proceso de desarrollo de software basado en el modelo de Proceso Unificado Racional(RUP) y enmarcado en las buenas prácticas Del Modelo de Madurez y Capacidad Integrado (CMMI) y en El ciclo de vida de Gestión de procesos de negocio (BPM).

Los resultados esta investigación se dividieron en cinco etapas, en la primera etapa se logran esclarecer las funciones de cada una de las KPA'S definidas por CMMI en su tercer nivel y en la categoría de Ingeniería. Esto permite conocer cuáles son las metas y prácticas indicadas por el modelo de calidad utilizado en la investigación y que a su vez permite definir procesos de negocio enmarcados en estas prácticas. En la segunda etapa se realizó el mapeo del flujo de trabajo de Implementación de RUP con cada una de las fases de CMMI para determinar cuáles de ellas están relacionadas con dicho flujo de trabajo. En la tercera etapa se realizo el diseño de las KPA's Solución Técnica, Verificación e Integración del Producto a través de procesos de negocio siguiendo los lineamientos indicados por cada una de las KPA's. En la cuarta etapa de la investigación se llevaron los procesos de negocio a una herramienta BPMS, donde se realizo el modelado de cada proceso usando notación BPMN2 y con la ejecución de los mismos en esta herramienta se observaron las ventajas que puede tener este tipo de herramienta en el seguimiento y monitoreo de procesos. Por último se presento el diseño y despliegue del proceso de Verificación usando JBPM5, este proceso sirvió como ejemplo para la ejecución de la actividad para cualquiera de los procesos definidos en la investigación. Como parte final se presentaran los tipos de indicadores que pueden ser obtenidos con la ejecución de estos procesos a través de un motor de procesos.

Este trabajo de investigación tiene entre uno de sus objetivos modelar los procesos de negocio que satisfacen las KPA de CMMI a través de BPM por lo cual este trabajo de Sivira, B (2011) sirve de guia para modelar el modelo que es uno de los objetivos de este trabajo de investigación y también se toma como referencia la parte de indicadores que se obtuvieron al final de la investigación.

Bases Teóricas

Proceso de desarrollo de Software

Según Sommerville (2005), un proceso de software es un conjunto de actividades y resultados asociados que producen un producto de software. Existen cuatro actividades fundamentales de procesos que son comunes para todos los procesos de software. Estas actividades son:

- Especificación del software: Es donde los clientes e ingenieros definen el software a producir y las restricciones sobre su operación.

- Desarrollo del software: donde el software se diseña y programa.

- Validación del software: donde el software se válida para asegurar que es lo que el cliente requiere.

- Evolución del Software: donde el software se modifica para adaptarlo a los cambios requeridos por el cliente y el mercado.

Por otra parte, están los métodos, que son un procedimiento para obtener algo, los ingenieros de software han creado diferentes métodos y modelos que guían técnicamente la construcción del software, según Sommerville (2005) en Sanchez (2011) “un método de ingeniería de software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable“

Modelo

Bennet (2007) afirma que como sucede en los mapas los modelos representan algo, son de utilidad en diferentes formas; así mismo los modelos tienen las siguientes características:

- Un modelo es más rápido y fácil de construir.
- Un modelo se puede usar en simulaciones, para aprender más sobre el ente al que representa.
- Un modelo puede evolucionar a medida que aprendemos más sobre una tarea o un problema.
- Podemos elegir que detalles queremos representar en un modelo y cuáles podemos ignorar. Un modelo es una abstracción.
- Un modelo puede representar elementos reales o imaginarios desde cualquier dominio.

Bennet (2007) también menciona que un modelo útil debe tener justo la cantidad adecuada de detalle y estructura y sólo debe representar lo que es importante para la tarea que se va a realizar. La mayoría de los modelos de desarrollo de sistemas utilizados en la actualidad tienen la forma de diagramas, que permiten el empleo de descripciones textuales y especificaciones lógicas o matemáticas de los procesos y los datos, Bennet (2007).

Ingeniería de Métodos

La ingeniería de métodos (IM) según Brinkkemper (1996) en Sanchez (2011) “es la disciplina de la ingeniería para diseñar, construir y adaptar métodos, técnicas y herramientas para el desarrollo de sistemas de información”.

La ingeniería de métodos establece una clara separación entre el producto que el método elabora y el proceso que elabora el producto. Según Harmsen (1997) en Sanchez (2011) el aspecto conceptual de un método consiste de un modelo de proceso y un conjunto de descripciones de producto. El modelo de procesos describe los pasos y actividades descritas por el método. Las descripciones del producto preescriben los contenidos del producto que será entregado por esos pasos o actividades.

Sanchez (2011), menciona que se puede decir que un modelo de producto es una especificación de un producto entregado o requerido dentro del método. Brinkkemper (1996) en Sánchez (2011) dice que, este modelo es la estructura de los productos, que pueden ser entregables, modelos, tablas, diagramas, documentos, etc que se producen en un método de desarrollo y el modelo de procesos es la descripción de los pasos que se realizan en el método, pueden ser determinar los requisitos, analizar los requisitos, diseñar modelo de datos, etc.

Métodos adaptativos (Agiles)

A partir de finales de 1990 y en la presente década, el proceso de software ha experimentado el surgimiento de modelos cada vez más adaptables. Estos modelos asumen que, con el correcto desarrollo, herramientas y prácticas era simplemente más rentable escribir el código rápidamente, ser evaluados por los clientes en el uso real, y refactorizar rápidamente en vez de tratar de anticipar y documentar todos los requisitos en el principio. De hecho, el número de métodos, incluyendo el desarrollo de Sistemas Dinámicos Método (DSDM), Feature-Driven Development (FDD), el desarrollo de software adaptable, Scrum, Extreme Programming (XP), Proceso Unificado Abierto(OpenUp), Agile RUP, Kanban, Métodos Lean, Crystal, y así sucesivamente, habla del constante

impulso de los procesos más eficaces y métodos más ligeros, Leffingwell (2011).

Metodologías Agiles

Según Canós, Letelier y Penadés (S.F), En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término ágil aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía ágil.

Según el Manifiesto se valora:

- **Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.** La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte

automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.

- **Desarrollar software que funciona más que conseguir una buena documentación.** La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental.

- **La colaboración con el cliente más que la negociación de un contrato.** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

- **Responder a los cambios más que seguir estrictamente un plan**. La habilidad de responder a los cambios que puedan surgir a los largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

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

1. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.

2. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.

3. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.

4. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.

5. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.

6. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.

7. El software que funciona es la medida principal de progreso.

8. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.

9. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.

10. La simplicidad es esencial.

11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.

12. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

En la Figura 2 podemos observar gráficamente la filosofía del método ágil comparado con los métodos tradicionales.

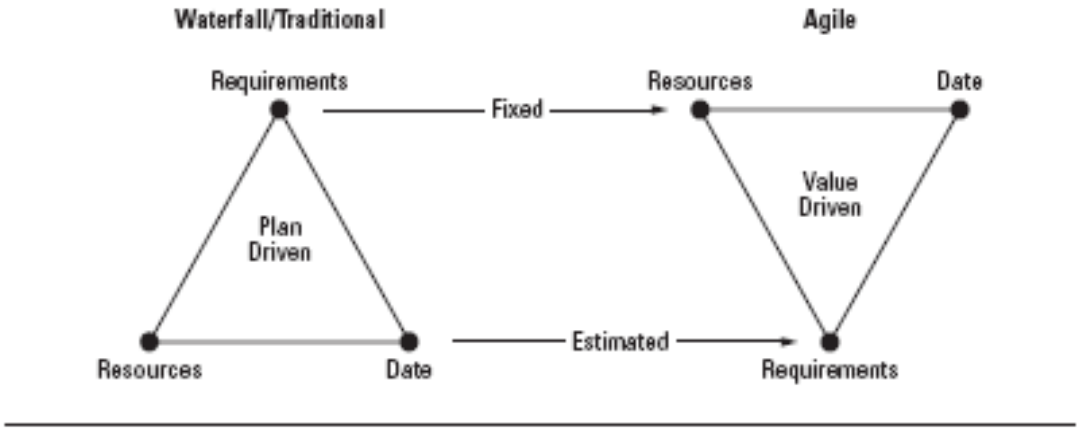


Figura 2. Método Tradicional Vs Ágil
Fuente: Leffingwell (2011).

Requisitos

Según Sawyer y Kotonya (2001) en Montilva (2007), " Un requisito es una propiedad que debe exhibir, cumplir o satisfacer un sistema desarrollado o adaptado para resolver un problema en particular"

Así mismo PFleeger (1998) en Montilva (2007) define requisito como "es un aspecto de un sistema o una descripción de aquello que el sistema es capáz de hacer a fin de cumplir su propósito".

Braude, (2003) en Montilva (2007) dice que "Los requisitos expresan qué se supone debe hacer una aplicación, no intentan expresar cómo lograr esas funciones".

Ingeniería de requisitos

Según Sommerville (2005), los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requisitos.

Según Pressman R. (2005), el proceso de ingeniería de requisitos puede ser descrito en 5 pasos distintos: Identificación de requisitos, Análisis de requisitos y negociación, especificación de requisitos, modelizado del sistema, validación de requisitos y gestión de requisitos.

Andriano N. (2006) afirma que la ingeniería de requisitos involucra el descubrir cuáles son las metas, necesidades y expectativas de los stakeholders, ajustar las expectativas de los mismos y comunicarlas a los desarrolladores. La ingeniería de requisitos está formada por una serie de procesos bien diferenciados como se muestra en la Figura 3.

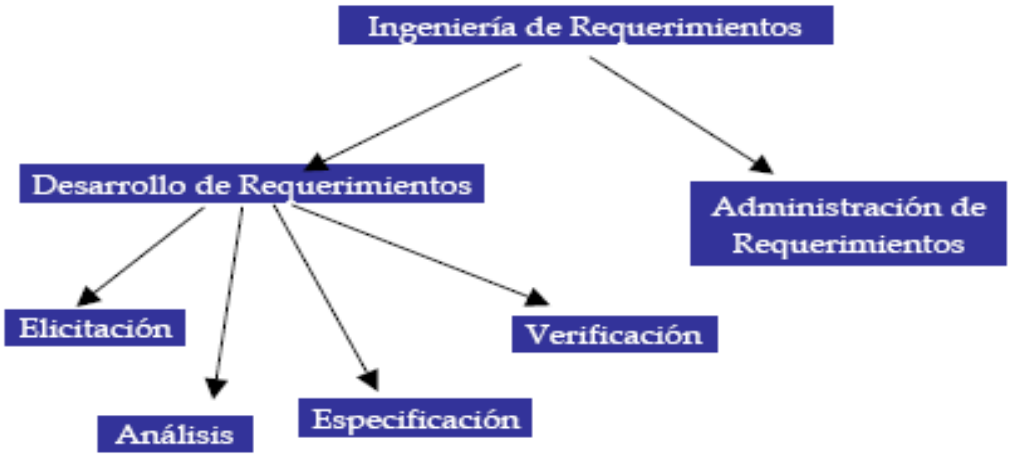


Figura 3. Procesos de la Ingeniería de requisitos.
Fuente: Andriano (2006).

Gestión de requisitos

Pressman R (2005) afirmó que los requisitos del sistema cambian y que el deseo de cambiar los requisitos persiste a lo largo de la vida del sistema. La gestión de requisitos es un conjunto de actividades que ayudan al equipo de trabajo a identificar, controlar y seguir los requisitos y los cambios en cualquier momento.

Sommerville I. (2005) expuso que la gestión de requisitos es el proceso de comprender y controlar los cambios en los requisitos del sistema y que era necesario mantenerse al tanto de los requisitos particulares y mantener vínculos entre los requisitos dependientes de forma que se pueda evaluar el impacto de los cambios en los requisitos.

En este mismo orden de ideas Borland (2005) citado en Andriano N. (2006) afirma que no es suficiente para las organizaciones recolectar requisitos desde múltiples stakeholders e incorporarlos independientemente

dentro de un sistema. Es necesario administrar estos de manera simultánea, además alguien, o algún equipo, debe ser responsable de administrar esos requisitos a lo largo del ciclo de vida de manera tal de mantener la visibilidad y control del proceso de entrega del software.

Así mismo Wiegers (1999) citado en Andriano N. (2006) dice que el acuerdo de los requisitos es el puente que une el desarrollo de los requisitos y la administración de los mismos. La administración de los requisitos incluye todas las actividades que mantienen la integridad y exactitud de los requisitos a medida que el proyecto progresa. Como se muestra en la Figura 4 la administración de los requisitos enfatiza:

- Control de los cambios a los requisitos que están sobre una línea base.

- Mantener los planes del proyecto actualizados de acuerdo con los requisitos.

- Control de versiones tanto de requisitos individuales como del documento de requisitos.

- Manejar las relaciones entre requisitos, y los links y dependencias entre los requisitos individuales y otros entregables del proyecto.

- Monitorear el estado de los requisitos sobre una línea base.

Figura 4. Actividades principales de la administración de requisitos. Fuente: Andriano (2006).

Montilva (2007), afirma que la gestión de requisitos se encarga de controlar los cambios a los requisitos establecidos en el documento de requisitos e involucra el manejo de cambios a los requisitos establecidos, el manejo de las relaciones entre requisitos, .el manejo de dependencias entre el documento de requisitos y los otros documentos producidos durante el desarrollo del sistema. En la figura 5 se muestra donde se ubica la gestión de requisitos dentro de la ingeniería de requisitos.

Figura 5. Gestión de Requisitos. Fuente: Montilva (2007).

Montilva (2007), menciona que el proceso de gestión de requisitos son los siguientes como se muestra en la figura 6:

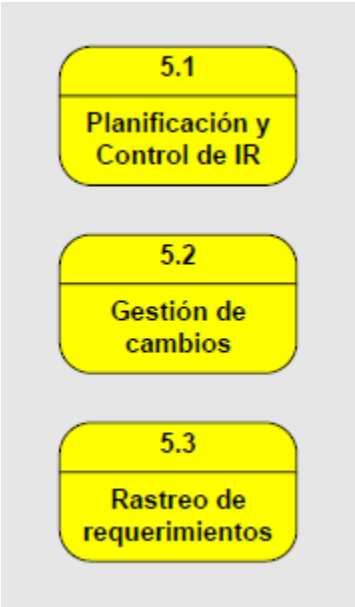


Figura 6. Procesos de Gestión de Requisitos. Fuente: Montilva (2007).

- Planificación y control de actividades de la ingeniería de requisitos:

En esta etapa se elabora el plan de ingeniería de requisitos, se estiman los costos y se determinan y seleccionan los recursos, por ejemplo las herramientas de apoyo a la ingeniería de requisitos

- Gestión de cambios:

Maneja los cambios en los requisitos una vez que el documento de requisitos ha sido liberado, en la figura 7 puede observarse el procesos de gestión de cambios.

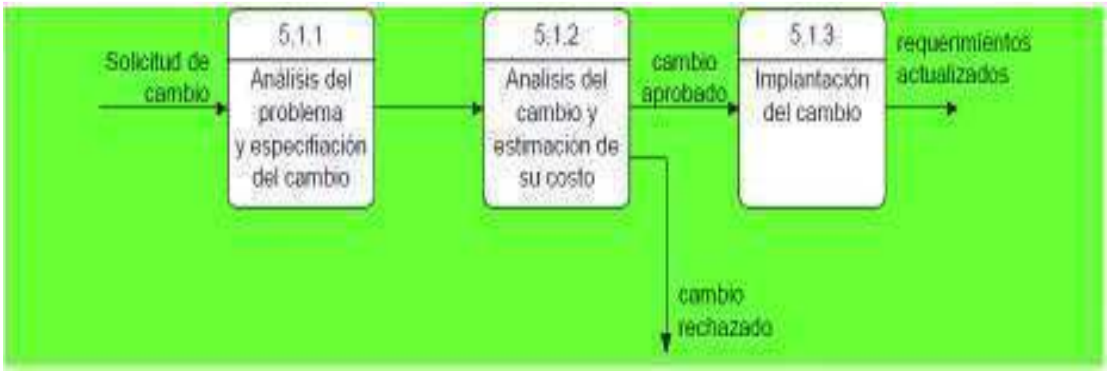


Figura 7. Procesos de gestión de cambios. Fuente: Montilva (2007).

- Rastreo de documentos:

Relaciona los requisitos con el diseño del sistema, sus componentes y las pruebas del sistema y asegura que todos los requisitos son implementados. Un requisito es rastreable sí se puede determinar quien propuso el requisito, porqué el requisito existe, que otros requisitos se relacionan con él y como el requisito se relaciona con los otros productos del desarrollo del sistema.

Montilva (2007), dice que las modalidades de rastreo (trazabilidad) son las siguientes:

- Rastreo desde atrás:

  Enlaza los requisitos a sus fuentes (personas o documentos).

- Rastreo desde adelante:

  Enlaza los requisitos con el diseño del sistema y los componentes implementados.

- Rastreo hacia atrás:

Enlaza el diseño del sistema y los componentes implementados con los requisitos.

- Rastreo hacia delante:

  Enlaza otros documentos que preceden al de requisitos con algunos requisitos que son relevantes.

En la figura 8 se muestra la matriz de rastreo y en la figura 9 la lista de rastreo.

Depends-on

	R1	R2	R3	R4	R5	R6
R1			*	*		
R2					*	*
R3				*	*	
R4		*				
R5						*
R6						

Figura 8. Matriz de rastreo. Fuente: Montilva (2007)

Requirement	Depends-on
R1	R3, R4
R2	R5, R6
R3	R4, R5
R4	R2
R5	R6

Figura 9. Lista de rastreo. Fuente: Montilva (2007)

La Ingeniería de requisitos en los métodos ágiles

- **Extreme programing (XP):**

Bernal, R (2007). Dice que xp está basado en los principios de simplicidad, comunicación, retroalimentación y valor. Trabaja con el equipo completo, usando prácticas simples, con la suficiente retroalimentación para permitirle saber siempre donde están.

XP no habla explícitamente de técnicas de requisitos en detalle, sino sobre el proceso general de desarrollo software y lo que se hace durante este proceso. Varias de las prácticas de XP (o de las técnicas usadas en estas prácticas) pueden ser comparadas con las técnicas de la Ingeniería de Requisitos (IR). Específicamente, durante el Planning Game, se usan técnicas de elicitación como las entrevistas, las tormentas de ideas y la priorización. XP usa las historias de usuarios (story cards) para la elicitación, Bernal, R (2007).

Antes de que las historias de usuario puedan ser escritas, los clientes tienen que pensar acerca de lo que ellos esperan que haga el sistema. Este proceso puede ser visto como una tormenta de ideas. Pensar acerca de una funcionalidad específica lleva a más ideas y a más historias de usuario. Cada historia se discute en forma de preguntas abiertas antes de la implementación. Inicialmente, los desarrolladores piden los detalles suficientes para ser capaces de estimar el esfuerzo de implementación de la historia. Basándose en estas estimaciones y en el tiempo disponible, los clientes priorizan las historias que tienen que ser resueltas en la siguiente iteración, Bernal, R (2007).

- **Modelado ágil (AM):**

La idea básica de AM es dar a los desarrolladores una guía de cómo construir modelos que ayuden a resolver los problemas de diseño. Como XP, AM apunta que los cambios son normales en el desarrollo software. AM no se refiere explícitamente a ninguna técnica de IR, pero alguna de las prácticas soportan varias técnicas de IR (como las pruebas y las tormentas de ideas). AM resalta la diferencia entre los modelos informales cuyo único propósito es ayudar a la comunicación cara a cara y los modelos que van a ser mantenidos como documentación del sistema. Estos últimos son los que con frecuencia se encuentran en las aproximaciones de la IR tradicional, Bernal, R (2007).

- **Scrum:**

Es un método para gestionar el proceso de desarrollo del sistema aplicando las ideas de flexibilidad, adaptabilidad y productividad provenientes de la teoría de control de procesos industriales. Scrum se enfoca en como el equipo de trabajo debería trabajar junto para producir trabajos de calidad en entornos cambiantes, Bernal, R (2007).

Las principales técnicas de Scrum son el backlog del producto, los sprints y las reuniones diarias. En relación con la IR el backlog juega un papel importante en Scrum. Todos los requisitos recolectados como necesarios o útiles para el producto se listan en el backlog del producto. Contiene una lista priorizada de todas las características, funciones, mejoras y arreglo de errores que deben ser implementados en el sistema. Este backlog puede

compararse con un documento de requisitos incompleto y cambiante, que contiene información necesaria para el desarrollo, Bernal, R (2007).

En cada sprint, de unos 30 días aproximadamente, las tareas más prioritarias son llevadas desde el backlog a la lista de tareas que tienen que ser desarrolladas en dicho sprint (sprint backlog). Durante el sprint, no se permiten cambios en el backlog, sin embargo hay flexibilidad absoluta para que cliente modifique el backlog para el siguiente sprint. Al finalizar el sprint se produce una reunión para demostrar la nueva funcionalidad al cliente y solicitar retroalimentación, Bernal, R (2007).

Los conocimientos adquiridos en esta reunión y el backlog actual sirven para planificar el siguiente sprint. En la reunión de revisión del sprint puede ser comparado con una revisión de los requisitos y una presentación de un prototipo evolucionado al cliente, Bernal, R (2007).

- **Crystal**

Las metodologías Crystal son una familia de diferentes metodologías desde las cuales se puede seleccionar la adecuada para un proyecto concreto. Los diferentes miembros de la familia pueden ser adaptadas para ajustarse a circunstancias variadas, Bernal, R (2007). Actualmente no están desarrollados todos los miembros de la familia Crystal. Algunas políticas estándar de todas las metodologías crystal pueden ser comparadas con las técnicas de IR:

- Desarrollo orientado a características(FDD)
- Método de desarrollo de sistemas dinámicos(DSDM):
- Desarrollo adaptativo del software (ASD)

Procesos de Negocio (BP)

Vázquez, González y León (s.f.), definen los procesos de negocio como colecciones de actividades interrelacionadas. Estas dan soporte a un producto o servicio, necesario para satisfacer las necesidades de un cliente. Su importancia en las organizaciones radica en que éstas se apoyan en la composición de actividades que presenta comúnmente problemas con el manejo de la información, el establecimiento de puntos de control para las actividades, la estandarización de procesos y la coexistencia de diferentes plataformas tecnológicas para soportarlos.

Manejo de Procesos de Negocio (BPM)

Según Garimella y otros (2008), Business Process Management (BPM) es un conjunto de métodos, herramientas y tecnologías utilizados para diseñar, representar, analizar y controlar procesos de negocio operacionales. BPM es un enfoque centrado en los procesos para mejorar el rendimiento que combina las tecnologías de la información con metodologías de proceso y gobierno. BPM es una colaboración entre personas de negocio y tecnólogos para fomentar procesos de negocio efectivos, ágiles y transparentes. BPM abarca personas, sistemas, funciones, negocios, clientes, proveedores y socios.

BPM combina métodos ya probados y establecidos de gestión de procesos con una nueva clase de herramientas de software empresarial. Ha posibilitado adelantos muy importantes en cuanto a la velocidad y agilidad con que las organizaciones mejoran el rendimiento de negocio. Con BPM:

- Los directores de negocio pueden, de forma más directa, medir, controlar y responder a todos los aspectos y elementos de sus procesos operacionales.

- Los directores de tecnologías de la información pueden aplicar sus habilidades y recursos de forma más directa en las operaciones de negocio.
- La dirección y los empleados de la organización pueden alinear mejor sus esfuerzos y mejorar la productividad y el rendimiento personal.
- La empresa, como un todo, puede responder de forma más rápida a cambios y desafíos a la hora de cumplir sus fines y objetivos.

Las Tres dimensiones de BPM

Según Garimella y otros (2008), BPM es llamado así acertadamente porque se dirige al extenso mundo de una compañía a través de sus tres dimensiones esenciales. En la figura 10 se muestran las tres dimensiones de BPM

Figura 10. Dimensiones que componen la notación BPMN.
Fuente: Garimella (2008)

- **El negocio: la dimensión de valor.**

La dimensión de negocio es la dimensión de valor y de la creación de valor tanto para los clientes como para los “stakeholders” (personas interesadas en la buena marcha de la empresa como empleados, accionistas y proveedores). BPM facilita directamente los fines y objetivos de negocio de la compañía: crecimiento sostenido de los ingresos brutos y mejora del rendimiento mínimo; aumento de la innovación; mejora de la productividad; incremento de la fidelidad y satisfacción del cliente y niveles elevados de eficiencia del personal.

BPM incorpora más capacidad que nunca para alinear actividades operacionales con objetivos y estrategias. Concentra los recursos y esfuerzos de la empresa en la creación de valor para el cliente. BPM también permite una respuesta mucho más rápida al cambio, fomentando la agilidad necesaria para la adaptación continua.

- **El proceso: la dimensión de transformación**

La dimensión de proceso crea valor a través de actividades estructuradas llamadas procesos. Los procesos operacionales transforman los recursos y materiales en productos o servicios para clientes y consumidores finales. Esta “transformación” es el modo en que funciona un negocio; el elixir mágico de la empresa. Mientras más efectiva sea esta transformación, con mayor éxito se crea valor.

Mediante BPM, los procesos de negocio son más efectivos, más transparentes y más ágiles. Los problemas se resuelven antes de que se conviertan en asuntos más delicados. Los procesos producen menos errores y estos se detectan más rápido y se resuelven antes.

Efectividad de los procesos: Los procesos efectivos son más coherentes, generan menos pérdidas y crean un valor neto mayor para clientes y “stakeholders”. BPM fomenta de forma directa un aumento en la efectividad de los procesos mediante la automatización adaptativa y la coordinación de personas, información y sistemas.

A diferencia de los métodos y las herramientas del pasado, BPM no impone la efectividad a través de sistemas de control rígidos e improductivos centrados en dominio s funcionales. En su lugar, BPM permite la respuesta y adaptación continuas a eventos y condiciones del mundo real y en tiempo real**.**

Transparencia de los procesos

La transparencia es la propiedad de apertura y visualización, y es crítica para la efectividad de las operaciones. Tiempo atrás, la transparencia eludía a las empresas, cuyos procesos están a menudo codificados en sistemas arcanos. Con BPM, se puede visualizar de forma directa todos los elementos del diseño de los procesos como el modelo, flujo de trabajo, reglas, sistemas y participantes así como su rendimiento en tiempo real, incluyendo eventos y tendencias. BPM permite a las personas de negocios gestionar de forma directa la estructura y flujo de los procesos y realizar el seguimiento de los resultados así como de las causas.

Agilidad en los procesos

De todas las demandas de las operaciones empresariales, quizás la más acuciante sea la necesidad de cambio, es decir, la capacidad de adaptación a eventos y circunstancias cambiantes manteniendo al mismo tiempo la

productividad y rendimiento globales. BPM proporciona agilidad en los procesos al minimizar el tiempo y el esfuerzo necesarios para traducir necesidades e ideas empresariales en acción. BPM permite a las personas de negocios definir procesos de forma rápida y precisa a través de los modelos de proceso. Les posibilita realizar análisis de futuro en escenarios empresariales. Les otorga derecho para configurar, personalizar y cambiar flujos de transacciones modificando las reglas de negocio. Directamente convierte diseños de procesos en ejecución, integrando sistemas y construyendo aplicaciones sin necesidad de código y sin fisuras. Además, cada plataforma BPM viene equipada con componentes tecnológicos que facilitan y aceleran el desarrollo sin código y la integración.

- **La gestión: la dimensión de capacitación**

La gestión es la dimensión de capacitación. La gestión pone a las personas y a los sistemas en movimiento y empuja a los procesos a la acción en pos de los fines y objetivos del negocio. Para la gestión, los procesos son las herramientas con las que se forja el éxito empresarial. Antes de BPM, construir y aplicar estas herramientas engendraba una mezcla poco manejable de automatización de clase empresarial, muchas herramientas de escritorio aisladas, métodos y técnicas manuales y fuerza bruta. Con BPM, se pueden tener todos los sistemas, métodos, herramientas y técnicas de desarrollo de procesos y la gestión de procesos en un sistema estructurado, completo, con la visibilidad y los controles necesarios para dirigirlo y afinarlo.

- **El catalizador: la tecnología BPM**

Líderes y directores de negocio conocen la función fundamental de los negocios, procesos y gestión de la empresa. Durante décadas, estos se han definido, estudiado y mejorado. La tecnología, sin embargo, ha evolucionado

más rápido y, recientemente, avances significativos han cambiado el juego. La tecnología BPM es el nuevo habilitador que ha llevado los negocios, procesos y la gestión a nuevos niveles. La tecnología BPM es el ingrediente clave de BPM, es el catalizador en una nueva creencia empresarial más rápida y más efectiva.

La tecnología BPM es el resultado de muchos años de experiencia en desarrollo y aplicación; el producto de los avances más actuales en sistemas y procesamiento de información; la cumbre de todas las arquitecturas, lenguajes y protocolos informáticos. La tecnología BPM constituye un gran avance, y un nuevo paradigma en cuanto a flexibilidad, gestión y control de información y datos. BPM, como práctica de gestión integral, es el resultado de la combinación de avances técnicos con métodos y prácticas establecidos, de un modelo empresarial centrado en el proceso.

La tecnología BPM incluye todo lo que necesita a la hora de diseñar, representar, analizar y controlar los procesos de negocio operacionales, entre los cuales podemos mencionar los siguientes:

El diseño y modelado de procesos: posibilitan que, de forma fácil y rigurosa se pueda definir procesos que abarcan cadenas de valor y coordinar los roles y comportamientos de todas las personas, sistemas y otros recursos necesarios.

La integración le permite incluir en los procesos de negocio cualquier sistema de información, sistema de control, fuente de datos o cualquier otra tecnología. La arquitectura orientada a servicios (SOA) lo hace más rápido y fácil que nunca. No es necesario desprenderse de las inversiones ya realizadas; todo se puede reutilizar.

Los entornos de trabajo de aplicaciones compuestas le permiten construir e implementar aplicaciones basadas en web casi de forma instantánea, completamente funcionales y sin necesidad de código.

La ejecución convierte de forma directa los modelos en acción en el mundo real, coordinando los procesos en tiempo real.

La supervisión de la actividad de negocio (BAM) realiza el seguimiento del rendimiento de los procesos mientras suceden, controlando muchos indicadores, mostrando las métricas de los procesos y tendencias clave y prediciendo futuros comportamientos.

El **control** le permite responder a eventos en los procesos de acuerdo a las circunstancias, como cambio en las reglas, notificaciones, excepciones y transferencia de incidentes a un nivel superior.

Ciclo de Vida de los Procesos de Negocio

Berrocal J, afirma que para la gestión de los procesos de negocio, de una organización, se proponen una serie de etapas y actividades que establecen el ciclo de vida, en la figura 11 puede observarse el ciclo de vida que se debe seguir para alcanzar, de una forma eficaz, todos los objetivos y beneficios perseguidos por BPM. Las principales fases son:

- Descubrimiento: el principal objetivo es descubrir y entender cada uno de los procesos de negocio que forman la organización. Especificando todos los detalles de cada uno de los requisitos y centrándose, principalmente, en las funcionalidades clave del sistema.

- Análisis: en esta fase se analizan cada uno de los procesos de negocio del sistema, modelándolos con las nuevas características y reglas que deben seguir para obtener una mayor productividad.

- Desarrollo: se desarrollan los procesos de negocio analizados y diseñados en la etapa anterior.

- Monitorizar: cada proceso de negocio debe ser medible para saber su grado de éxito y calidad con el que ha sido llevado a cabo; de esta forma, se pueden analizar los resultados de cada uno de los procesos para que puedan ser redefinidos y optimizados.

Figura 11. Ciclo de vida iterativo de los procesos de negocio. Fuente: Berrocal J.

- Optimizar: aquellos procesos que no han cumplido las expectativas deseadas, bien porque no poseen un conjunto coherente de tareas, o bien porque las necesidades han cambiado, son optimizados para que puedan mejorar su rendimiento y así también el de la empresa. Si se

necesita crear un nuevo software que soporte las optimizaciones, será imprescindible que estos procesos pasen, de nuevo, a la fase de análisis.

BPMN

White, S (2009) en Bazán P (2009), define a BPMN como un diagrama de procesos de negocio (BPD) basado en una técnica adaptada de diagramas de flujo para la creación de modelos gráfico de operaciones de procesos de negocio. La importancia de BPMN se debe a que el mundo de los negocios ha cambiado drásticamente en los últimos años. Los procesos son coordinados dentro de los límites naturales de las organizaciones y a su vez pueden interactuar con procesos de otras organizaciones. Actualmente un proceso de negocio abarca múltiples participantes y la coordinación puede ser compleja.

Hasta la aparición de BPMN no existía un estándar sobre técnicas de modelado desarrollado para estos fines. BPMN ha sido desarrollado para proveer a los usuarios de una notación estándar de forma análoga a como UML estandarizó el mundo de la ingeniería del software. BPMN está dirigido a los analistas de negocio, arquitectos de sistemas e ingenieros de software. Fue desarrollado para mejorar todo el ciclo de vida del desarrollo de procesos desde el diseño de los mismos.

BPMN define un único tipo de diagrama que posee múltiples vistas derivadas del mismo meta-modelo de ejecución del proceso. Como resultado, la implementación es una vista lógica del proceso en un lenguaje de ejecución de procesos de negocio (BPML).

Elementos del lenguaje

White, S (2009) en Bazan (2009) menciona que un diagrama de proceso de negocio está compuesto por un conjunto de elementos gráficos. Los elementos utilizados para construir los diagramas fueron elegidos para ser distinguibles unos de otros y utilizar las figuras que son familiares a la mayoría de los diseñadores. Por ejemplo, las actividades se representan mediante rectángulos y las decisiones mediante rombos. Cabe destacar que uno de los objetivos del desarrollo de BPMN fue crear un mecanismo sencillo para la creación de modelos de procesos de negocio y al mismo tiempo ser capaz de manejar la complejidad inherente de los mismos. El enfoque adoptado para manejar estos dos requisitos fue la organización de la gráfica de los aspectos de la notación en categorías específicas. Éste provee un pequeño conjunto de categorías de notación que permite al lector del diagrama de procesos de negocio reconocer fácilmente los elementos básicos y comprender el diagrama. Dentro de las categorías básicas de elementos se pueden incluir variaciones adicionales o información para soportar requisitos complejos sin tener un cambio drástico en la mirada y sentido básico del diagrama.

Las cuatro categorías básicas de elementos son:

- Flow objects (objetos que representan el flujo).
- Connecting objects (objetos conectores).
- Swinlanes (andariveles)
- Artefacts (artefactos)

y se encuentran representados en la figura 12

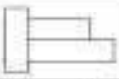
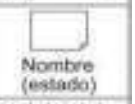
Categoría	Elemento	Descripción	Gráfica
Flujos	Evento	Es algo que sucede durante el curso del proceso de negocio. Afectan al flujo del proceso. Normalmente tienen una causa (disparador) o un impacto (resultado). Dependiendo de cuando afectan al flujo serán eventos iniciales, intermedios o finales.	
	Actividad	Es un termino genérico para el trabajo que realiza una compañía. Puede ser atómica (tarea) o compuesta (sub-proceso). Para indicar la no atomicidad se coloca un signo + en la esquina del símbolo de actividad.	
	Gateway	Se utiliza para controla la convergencia o divergencia de flujos. Representa una decisión para mezclar o unir caminos.	
Conexiones	Flujo de secuencia	Se utiliza para mostrar el orden o secuencia en que las actividades se realizan en un proceso	
	Flujo de mensajes	Se utiliza para mostrar el flujo de mensajes entre dos participantes separados.	
	Asociación	Se utiliza para mostrar entradas y salidas de actividades.	
Swimlanes	Pool (fondo común	Representa un participante en un proceso. Actúa como contenedor grafico para particionar un conjunto de actividades.	
	Lane (sendero)	Es una sub-particion dentro de un pool y puede extenderse a todo lo largo o ancho del pool. Se utilizan para organizar y categorizar actividades.	
Artefactos	Objetos de datos	Mecanismo para mostrar como los datos son requeridos y producidos por las actividades. Se conectan a las actividades por asociaciones.	
	Grupos	Se utiliza para documentación o para propósitos de análisis, pero no afecta al Flujo de Secuencias	
	Anotaciones	Mecanismo para que quien esta modelando provea información adicional para el lector del diagrama.	

Figura 12. Elementos notacionales de BPM. Fuente: Bazán (2009)

Definición de términos básicos

BPM: Se llama gestión de procesos de negocio a la metodología corporativa cuyo objetivo es mejorar la eficiencia a través de la gestión de los procesos de negocio.

BPMS: Son un conjunto de herramientas que dan soporte para cumplir con el ciclo de vida de BPM.

Gestores: Los gestores de software son los responsables de la planificación del desarrollo de los proyectos. Supervisan el trabajo para asegurar que se lleva a cabo conforme los estándares requeridos y supervisan el progreso para comprobar que el desarrollo se ajusta al tiempo previsto y al presupuesto.

Método: Modo ordenado y sistemático de proceder para llegar a un resultado o fin determinado.

Métodos ágiles: Métodos de desarrollo de software dirigidos a la entrega rápida del mismo. El software se desarrolla y se entrega en incrementos, y se minimiza el proceso de documentación y la burocracia.

Sistemas IT: Son los sistemas de información y tecnología.

Stakeholders: Este término se utiliza para referirse a cualquier persona o grupo que se verá afectado por el sistema y todos aquellos en la organización que se pueden ver afectados por su instalación.

CAPITULO III

MARCO METODOLÓGICO

El presente estudio propone el desarrollo de un modelo de mejora continua para la gestión de requisitos de software usando métodos ágiles; de acuerdo a los objetivos planteados se ha establecido la naturaleza del estudio y las diferentes fases que harán factible el cumplimiento de su objetivo integral; el presente capítulo desarrolla aspectos metodológicos relativos al tipo de estudio y su diseño de investigación.

Naturaleza del Estudio

Este trabajo es una investigación de campo tipo descriptiva, es de campo, porque se siguen las estrategias basadas en métodos que permiten conocer los datos en forma directa de la realidad. Constituye un proceso sistemático riguroso y racional de recolección. Igualmente, es descriptiva, ya que comprende además de la descripción, el registro, análisis e interpretación de la naturaleza actual del problema del fenómeno estudiado, Ballestrini (2002).

Se entiende por investigación de campo, el análisis sistemático de problemas en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y sus efectos, o predecir su ocurrencia, haciendo uso de métodos característicos de cualquiera de los paradigmas o enfoques de investigación

conocidos o en desarrollo. Los datos de interés son recogidos en forma directa de la realidad; en este sentido se trata de investigaciones a partir de datos originales o primarios. Sin embargo, se aceptan también estudios sobre datos censales o muéstrales no recogidos por el estudiante, siempre y cuando se utilicen los registros originales con los datos no agregados; o cuando se trate de estudios que impliquen la construcción o uso de series históricas y, en general, la recolección y organización de datos publicados para su análisis mediante procedimientos estadísticos, modelos matemáticos, o de otro tipo, UPEL (2010).

Esta definición se adapta al trabajo de investigación desarrollado para él, ya que el diseño de un Modelo de mejora continua para la gestión de requisitos de software usando métodos ágiles, requiere que el investigador lleve a cabo la recolección de la información por revisión bibliográfica.

Para esta investigación se plantea como un proyecto factible, según Manual de trabajos de grado UPEL (2010), el proyecto factible consiste en la investigación, elaboración y desarrollo de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El proyecto debe tener apoyo en una investigación de tipo documental, de campo o un diseño que incluya ambas modalidades.

El proyecto factible comprende las siguientes etapas generales: diagnóstico, planteamiento y fundamentación teórica de la propuesta; procedimiento metodológico, actividades y recursos necesarios para su ejecución, análisis y conclusiones sobre viabilidad y realización del proyecto; y en caso de su desarrollo, la ejecución de la propuesta y la evaluación tanto del proceso como de sus resultados, Manual de trabajos de grado UPEL (2010).

Diseño de Investigación

Según Ballestrini (2006), un diseño de investigación se define como el plan global de investigación que integra de un modo coherente y adecuado técnicas que recogen datos a utilizar, análisis previstos y objetivos; el diseño de una investigación intenta dar de una manera clara y no ambigua respuestas a las preguntas planteadas en la misma.

En este trabajo de escogió un diseño no experimental, el cual Ballestrini (2006) menciona que existen muchas propuestas de clasificación de los tipos de diseño de investigación, pero de manera primaria, en relación al tipo de datos que se deben recolectar, estos se pueden clasificar en diseños de campo y diseños bibliográficos. Sin embargo, es posible situar dentro de los diseños de campo, otra clasificación, los no experimentales, en el cual se ubican los estudios exploratorios, descriptivos, diagnósticos, evaluativos, los causales y los proyectos factibles, donde se observan los hechos tal como se manifiestan en su ambiente natural, y en este sentido, no se manipulan de manera intencional las variables.

La presente investigación se enmarca en la modalidad de estudio de proyecto factible, el cual según Ucla (2002), consiste en una proposición sustentada en un modelo viable para resolver un problema práctico planteado, tendente a satisfacer las necesidades institucionales o sociales y pueden referirse a la formulación de políticas, programas, tecnologías, métodos y procesos. En relación a esto el autor busca dar respuesta a la necesidad de las empresas de desarrollo de software, como lo es el diseño de un modelo de mejora continua para la gestión de requisitos. Para ello, se propone un modelo de mejora continua para la gestión de requisitos usando métodos ágiles, lo cual justifica la adopción de este tipo de estudio en el presente proyecto.

Siguiendo las características de una investigación descriptiva, los pasos a seguir en el trabajo son: (a) Investigación teórica y documental, (b) Definición de la propuesta sobre un caso de estudio y (c) Conclusiones y recomendaciones.

Investigación Teórica y documental

Esta fase de la metodología permite construir las bases teóricas de la investigación, a través de la consulta de libros, referencias electrónicas, trabajos de grado y artículos científicos certificados. A partir de la información recolectada en estos medios, se conformó el Capítulo II de este trabajo.

Definición de la propuesta sobre un caso de estudio

Siendo un proyecto factible la investigación tiene tres (3) fases de desarrollo las cuales se muestran a continuación:

Fase I. Estudio Diagnóstico

Comenzando con la información obtenida de la revisión bibliográfica de distintos artículos, trabajos de grado y tesis, se realizará el análisis de la misma y será usada como información para la elaboración del modelo de gestión de requisitos que se plantea en la siguiente investigación.

La realización de esta fase permitirá conocer los métodos y modelos que se usan actualmente en las empresas de desarrollo de software, lo que permitirá conocer las fortalezas y debilidades de los mismos.

Fase II. Estudio de Factibilidad

En esta fase se especificara la posibilidad de elaborar la propuesta siguiendo un conjunto de criterios descritos a continuación:

Factibilidad Social

Esta propuesta presentará una solución de gran aporte para gestionar requisitos de software en las empresas de desarrollo de software que necesiten utilizar métodos ágiles para su proceso de gestión de requisitos. Este tipo de estudio proporcionara un aporte muy importante en la parte de Ingeniería de requisitos de los procesos de desarrollo de software usados en las organizaciones.

La propuesta puede dar mucho valor en los tiempos de respuesta y en la gestión de los requisitos del software a construir, trayendo consigo la tan ansiada satisfacción de los clientes que es uno de los grandes problemas con los que se encuentran las empresas donde se desarrolla software.

Es por lo que se expuso anteriormente que la solución se considera factible desde el punto de vista social.

Factibilidad Tecnológica

Para el modelado del modelo se requiere de una herramienta que facilite realizar esta tarea. En el mercado hoy en día existen muchas herramientas BPMS en licencia libre o propietaria que pueden ser usadas para modelar el modelo propuesto.

Los costos relacionados con la herramienta de software a usar para la automatización son gratis, ya que en esta investigación es posible usar una herramienta de software libre, las mismas se encuentran disponibles a cualquier usuario y sin la necesidad de pagar ningún licenciamiento por la misma.

En cuanto al hardware a usar, la automatización es viable realizarla en equipos personales de especificaciones no tan complejas ni de tan alto costo

para las empresas. De todas formas la empresa usada como caso de estudio ya cuenta con equipos donde puede realizarse las pruebas necesarias.

Se puede decir que la propuesta es factible técnicamente porque el software requerido es de licencia libre y los equipos de hardware son accesibles para cualquier empresa que desarrolle software y para el caso del caso de estudio ya se cuenta con dichos equipos de hardware.

Factibilidad Operativa

Al momento de poner en uso la propuesta, al conocer el modelo las personas responsables pueden usar sin mayor problema este modelo y la herramienta BPMS.

Por lo tanto, se garantiza la aplicación del modelo desde el punto de vista operativo.

Factibilidad Económica

En el desarrollo de la propuesta no se necesita gasto ya que se cuenta con las herramientas necesarias para su implementación, además que la licencia de la herramienta BPMS para la automatización de los procesos es gratuita.

Factibilidad Institucional

La propuesta genera muchos beneficios en las empresas, porque mejora su proceso de ingeniería de requisitos así como ayuda a agilizar este proceso de la ingeniería de software, permitiendo gestionar requisitos de forma ágil y de una manera favorable para el desarrollo de mismo.

Fase 3: Diseño de Propuesta

Con los resultados obtenidos en el diagnóstico y en el estudio de factibilidad, se procede al diseño de la propuesta. La investigación está dividida en cuatro (4) etapas las cuales se mencionan a continuación:

Etapa I: Estudio de la metodología ágil Scrum

En esta etapa se realizará el estudio de las prácticas usadas por la metodología scrum para cumplir con la etapa de requisitos del proceso de desarrollo de software, de donde se extraerán las características más resaltante de este en la gestión de requisitos, analizando las fortalezas mas resaltantes para generar un modelo con las mejores prácticas.

Etapa II: Diseñar el modelo de mejora continua para la gestión de requisitos

Con la información recopilada en las etapas anteriores se procederá al diseño del modelo para gestionar requisitos.

Etapa III: Modelado del modelo propuesto bajo un enfoque BPM

El diseño del modelo realizado en la etapa anterior será modelado en una herramienta BPMS.

Etapa IV: Diseñar indicadores de desempeño del modelo propuesto.

Por último se diseñaran indicadores de desempeño del modelo propuesto, los cuales permitirán identificar posibles fallos o mejoras del mismo y servir de base para la realización del sistema de mejora continua.

CAPITULO IV

RESULTADO

En los capítulos anteriores se mostro una visión general de los objetivos que se quieren lograr a través de este estudio y se estudiaron los antecedentes y el marco teórico relacionado, definiéndose las fases de estudio que serán desarrolladas en este capítulo.

Un modelo de mejora continua para la gestión de requisitos permite a las empresas cumplir con uno de los pasos fundamentales que se requieren en todo proceso de desarrollo de software para darle una probabilidad mayor de éxito a los proyectos que se pretendan desarrollar. La presente investigación propone un modelo de mejora continua para la gestión de requisitos de software apoyado en la metodología ágil Scrum y modelado en una herramienta bpms usando una filosofía BPM.

Este trabajo está dividido en cuatro (4) fases de trabajo que comprenden desde el estudio de la metodología ágil Scrum, la definición conceptual del modelo de mejora continua usando el enfoque de ingeniería de métodos, el ciclo BPM y las actividades de gestión de requisitos establecidas en Scrum, construcción del modelo de procesos en notación BPMN 2.0 adaptado a la especificación funcional definida y construir un sistema de indicadores de desempeño para el modelo de mejora continua.

Etapa I

Estudio de la metodología ágil Scrum

En el segundo capítulo de esta investigación se describen las características más relevantes de las metodologías ágiles; en esta etapa se profundizara en la metodología ágil Scrum; esta metodología ágil define un conjunto de actividades para construir software, en este sentido establece algunas orientadas a la gestión de requisitos, las cuales serán estudiadas en este capítulo.

Ayerbe R.(2007), define a Scrum como un método para gestionar el proceso de desarrollo del sistema aplicando las ideas de flexibilidad, adaptabilidad y productividad provenientes de la teoría de control de procesos industriales. Las principales técnicas de Scrum son el backlog del producto, los sprints y las reuniones diarias.

Palacio (2009), dice que la estructura ágil de Scrum está compuesta de cinco etapas como se muestra en la figura 13:

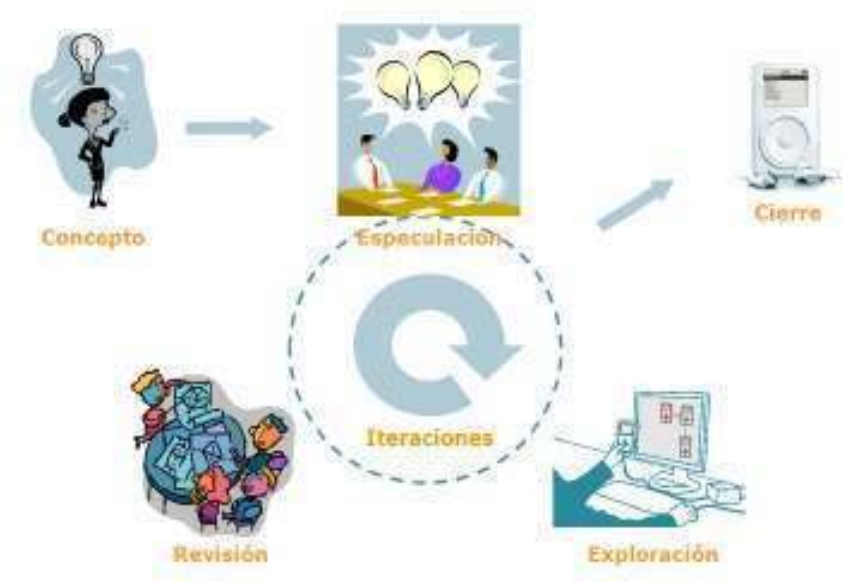


Figura 13. Etapas de Scrum. Fuente: Palacio (2009)

- Concepto: Es donde se crea la visión del producto o servicio que se quiere obtener, y se decide y selecciona las personas que lo llevaran a cabo.

- Especulación: En esta etapa se determinan las limitaciones impuestas por el entorno de negocio (costes y agendas principalmente) y se determina la primera aproximación de lo que se puede producir.

- Exploración: Desarrollo de las funcionalidades que para generar el siguiente incremento de producto, ha determinado el equipo en la etapa anterior.

- Revisión: El equipo y los usuarios revisan las funcionalidades construidas hasta ese momento. Trabajan y operan con el producto real para determinar su alineación y dirección con el objetivo.

- Cierre: Al llegar a la fecha de entrega de una versión de producto (fijada en la fase de concepto y revisada en las diferentes fases de especulación), se obtiene el producto esperado.

Ciclo de vida de Scrum

Fernández, Y (2009) muestra en la figura 14 el ciclo de vida propuesto por la metodología Scrum, en donde se diferencian las siguientes fases:

- Fase de Pre-game: Durante esta fase los clientes y los miembros del equipo Scrum definen las historias de usuario. Las historias de usuario son priorizadas por su importancia, según el cliente, y refinadas para la identificación de las tareas específicas a desarrollar. El resultado de esta fase es el backlog del producto. Finalmente las historias de usuario son agrupadas en sprints de corta duración (aproximadamente 30 días) según su prioridad.

- Fase de Planificación: Por cada sprint, se realiza una reunión de planificación que establece las historias de usuario que deben ser implementadas y la fecha de finalización de la misma. La estimación de las historias de usuario se realizan en forma conjunta entre clientes, jefe de proyecto y desarrolladores utilizando la técnica de planning game. El objetivo es mover aquellas historias de usuario con mayor prioridad para el cliente al backlog del sprint de forma que no

puedan producirse cambios sobre los aspectos que se encuentran en fase de desarrollo durante el sprint.

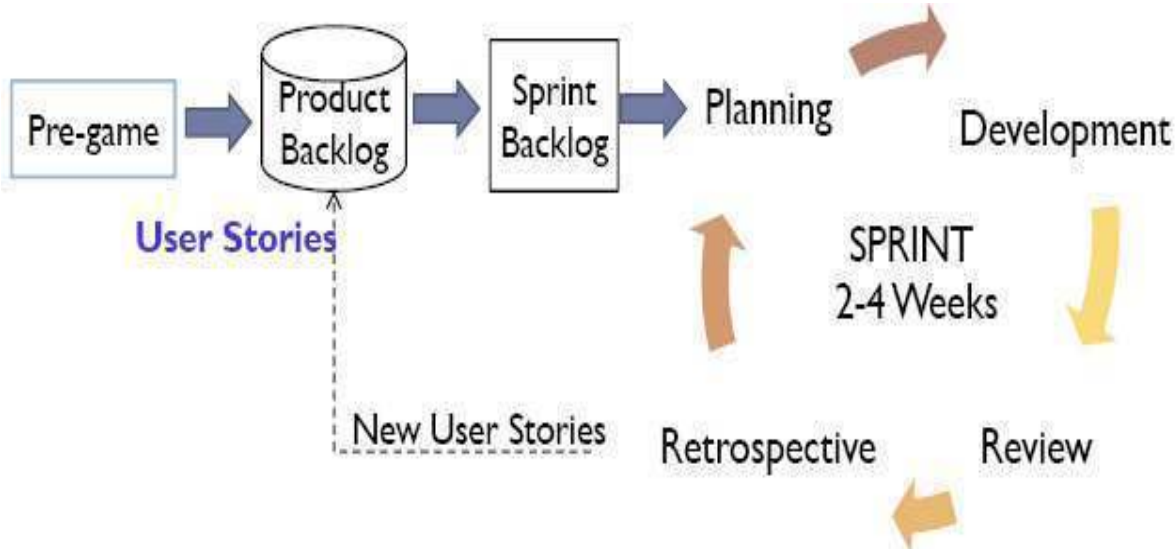


Figura 14. Modelo de desarrollo Scrum. Fuente: Fernández Y (2009)

- Reunión diaria: Durante cada día de que consta el sprint se realiza una reunión operativa, informal y ágil con el equipo de desarrollo, de un máximo de quince minutos, en la que a cada integrante del equipo se le hacen tres preguntas:

  ¿Qué tareas ha hecho desde la última reunión?
  ¿Qué tareas va ha hacer hoy?
  ¿Qué obstáculos ha identificado que puedan impedir su avance normal?

- Fase de revisión: El resultado de un sprint podría ser un entregable, un documento, parte del producto final, o algún otro artefacto que demuestre los avances realizados en el sprint. En la reunión de revisión del proyecto se muestran estos avances y se incorporan, si fuese necesario, nuevas historias de usuario al backlog del producto. Se considera que una historia de usuario está 100% completa si

supera los test unitarios, las pruebas de aceptación, test de calidad, el código está construido e integrado satisfactoriamente, está basado en un diseño factorizado sin duplicaciones, el código es claro, estructurado y autodocumentado y, por supuesto, es aceptado por el cliente.

- Fase de retrospección o análisis: La reunión retrospectiva permite mejorar de forma continua el proceso de desarrollo: el equipo de desarrollo analizará los objetivos marcados inicialmente en el backlog del sprint, las capacidades o habilidades del equipo, las condiciones de negocio actuales, los avances tecnológicos, los aspectos positivos y negativos del sprint, etc. Todo ello servirá de retroalimentación para aplicar los cambios y ajustes necesarios para el siguiente sprint.

Fernández Y (2009), afirma que La adopción de prácticas ágiles como la planificación por sprints, reuniones diarias, y backlogs de productos han demostrado una mejora de la comunicación en equipos de desarrollo y mejora también en la gestión de requisitos.

Requisitos con Scrum

Palacio (2008), muestra que Scrum para software emplea dos formatos para el registro y comunicación de los requisitos:

- Product Backlog
- Sprint Backlog

El product Backlog se sitúa en el área de requisitos o necesidades de negocio desde el punto de vista del cliente. En relación con la ingeniería de requisitos el backlog juega un papel importante en Scrum; todos los

requisitos recolectados como necesarios o útiles para el producto se listan en el backlog del producto, en la figura 15 se puede observar el ciclo del backlog del producto, Palacio (2008).

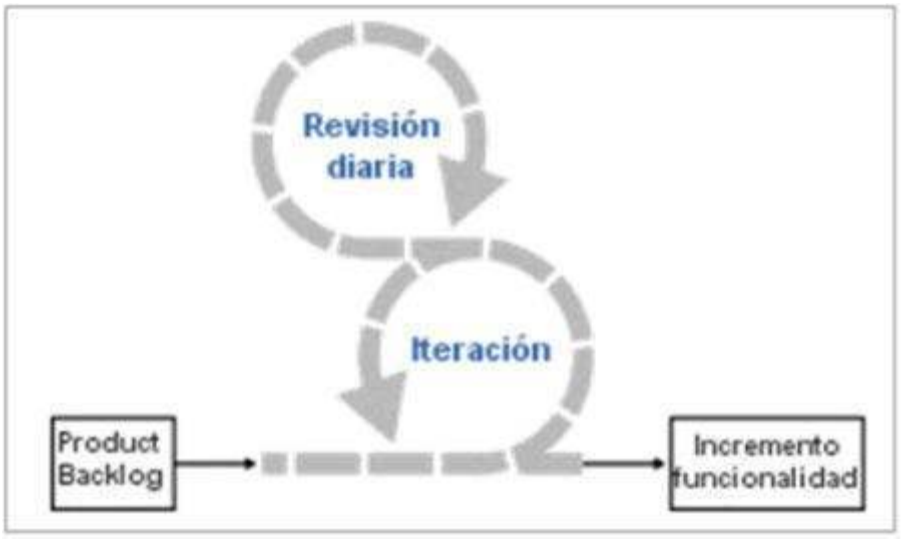


Figura 15. Ciclo del Backlog del producto. Fuente: Internet
http://www.navegapolis.net/content/view/801/62/

El Product Backlog se sitúa en el área de especificación de los requisitos de software necesarios para dar respuesta a las funcionalidades esperadas por el cliente.

Product Backlog: Los requisitos del Cliente

El product backlog es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de las sucesivas iteraciones de desarrollo, Palacio (2008).

El product backlog representa todo aquello que esperan los clientes, usuarios, y en general los interesados en el producto. Todo lo que suponga

un trabajo que debe realizar el equipo tiene que estar reflejado en el backlog, Palacio (2008). En la figura 16 se puede observar un ejemplo de product backlog.

Id	Módulo	Descripción	Est.	Por
Crítico				
1		Plataforma tecnológica	30	AR
2	Cliente	Interfaz de usuario	40	LR
3	Cliente	Un usuario se registra en el sistema	40	LR
4	Trastienda	El operador define el flujo y textos de un expediente	60	AR
5	Trastienda	Etc...	999.	XX
Necesario				
6	Cliente	El usuario modifica su ficha personal	30	AR
7	Cliente	El usuario consulta los expedientes asignados	15	LR
8	Cliente	El usuario tramita un expediente	35	LR

Figura 16 – Ejemplo del Product Backlog de Scrum. Fuente: Fernández (2009)

Estos son algunos ejemplos de posibles entradas de un backlog:

- Permitir a los usuarios la consulta de las obras publicadas por un determinado autor.
- Reducir el tiempo de instalación del programa.

- Mejorar la escalabilidad del sistema.
- Permitir la consulta de una obra a través de un API web.

Palacio (2008), afirma que a diferencia de un documento de requisitos del sistema, el product backlog nunca se da por completo; está en continuo crecimiento y evolución. Habitualmente se comienza a elaborar con el resultado de una reunión de brainstorming donde colabora todo el equipo partiendo de la visión del propietario del producto.

El product backlog evolucionará de forma continua mientras el producto esté en el mercado, para darle valor de forma continua y mantenerlo útil y competitivo, Palacio (2008).

Para comenzar el desarrollo se necesita una visión de los objetivos que se quieren conseguir con el producto, comprendida y conocida por todo el equipo, y elementos suficientes en el product backlog para llevar a cabo el primer sprint, Palacio (2008).

Formato del product backlog

Palacio (2008), dice que es recomendable que el formato incluya la siguiente información en cada línea:

- Identificador único de la funcionalidad o trabajo.
- Descripción de la funcionalidad.
- Campo o sistema de priorización.
- Estimación.

Dependiendo del tipo de proyecto, funcionamiento del equipo y la organización, pueden resultar aconsejables otros campos:

- Observaciones.
- Criterio de validación.
- Persona asignada
- Número de sprint en el que se realiza
- Módulo del sistema al que pertenece.

Sprint Backlog

El sprint backlog es la lista que descompone las funcionalidades del product backlog en las tareas necesarias para construir un incremento: una parte completa y operativa del producto, Palacio (2008). En la figura 17 de puede observar un ejemplo de Sprint Backlog.

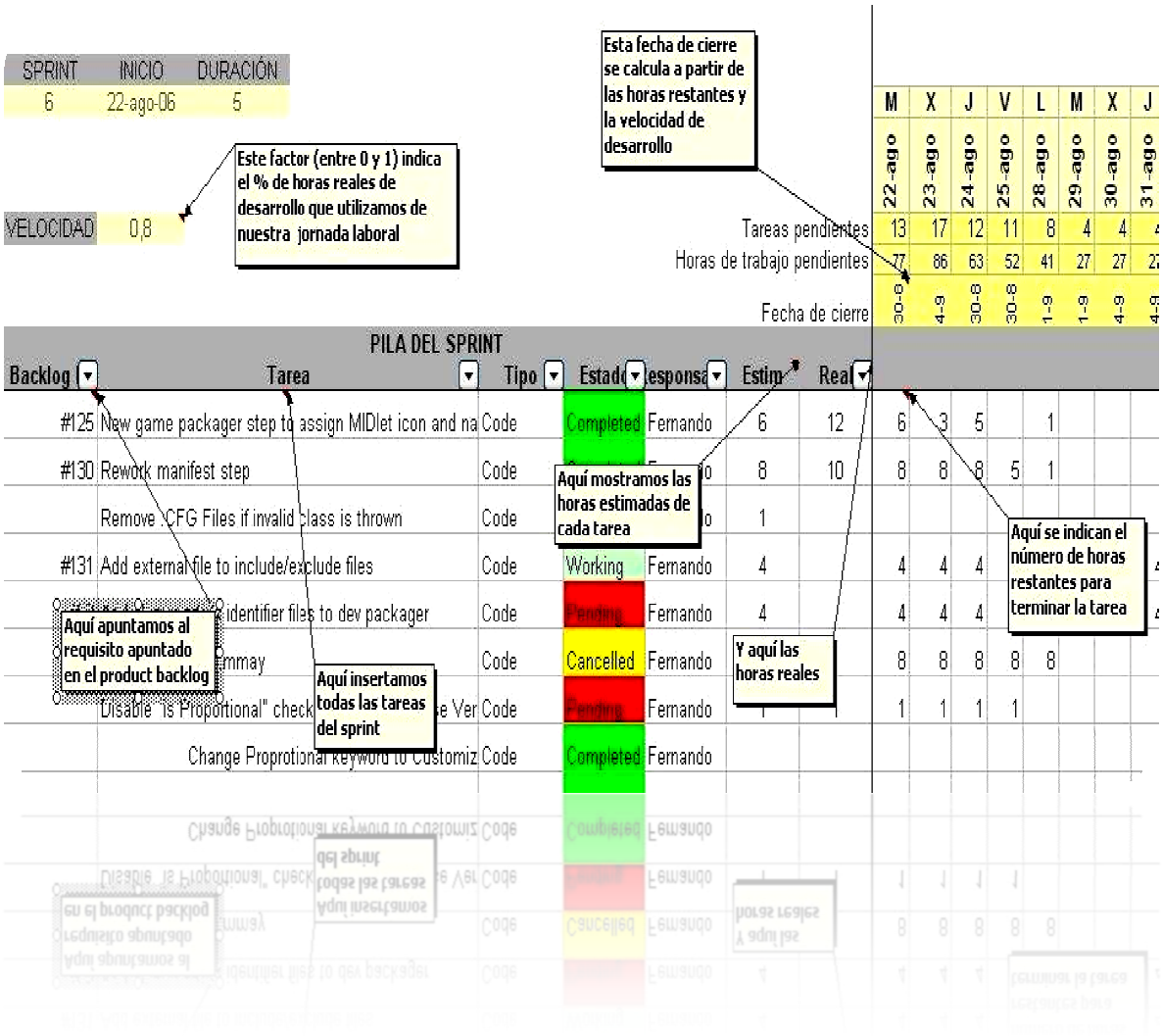


Figura 17. Sprint Backlog. Fuente: Palacio (2008).

En el sprint backlog se asigna a cada tarea la persona que la va a llevar a cabo, y se indica el tiempo de trabajo que se estima para terminarla, Palacio (2009).

Es útil porque descompone el proyecto en tareas de tamaño adecuado para determinar el avance a diario; e identificar riesgos y problemas sin necesidad de procesos complejos de gestión, así como también es una

herramienta de soporte para la comunicación directa del equipo, Palacio (2009).

Las condiciones para el Sprint Backlog son las siguientes:

- Realizado de forma conjunta por todos los miembros del equipo.
- Cubre todas las tareas identificadas por el equipo para conseguir el objetivo del sprint.
- Sólo el equipo lo puede modificar durante el sprint.
- El tamaño de cada tarea esta en un rango de 4 a 16 horas de trabajo.
- Es visible para todo el equipo. Idealmente en una pizarra o pared en el mismo espacio físico donde trabaja el equipo.

  Para dar formato y soporte al sprint backlog se puede usar alguna de las siguientes opciones:

  - Hoja de cálculo.
  - Pizarra o pared física.
  - Herramienta colaborativa o de gestión de proyectos.

Roles o responsabilidades con Scrum

1. Responsabilidad del Producto (Propietario del producto):

En el proyecto hay una persona conocedora del entorno de negocio del cliente y de la visión del producto. Representa a todos los interesados en el producto final y es el responsable del Produck Backlog.

Se le suele denominar propietario del producto y es el responsable de obtener el resultado de mayor valor posible para los usuarios o clientes. Es el responsable de decidir cómo debe ser el resultado final; en desarrollos internos puede ser el product manager o el responsable de marketing quien asume este rol, En desarrollos para clientes externos lo más aconsejable es que sea el responsable del proceso de adquisición del cliente.

2. Responsabilidad del desarrollo (El Equipo)

Todo el equipo de desarrollo, incluido el propietario del producto conoce la metodología Scrum, y son los auténticos responsables del resultado. Es un equipo multidisciplinario que cubre todas las habilidades necesarias para generar el resultado. Se auto-gestiona y auto-organiza, y dispone de atribuciones suficientes en la organización para tomar decisiones sobre cómo realizar su trabajo.

3. Responsabilidad del funcionamiento de Scrum (scrum manager)

La organización debe garantizar el funcionamiento de los procesos y metodologías que emplea, En el modelo de Scrum, esta responsabilidad se garantiza integrando en el equipo una persona con el rol de ScrumMaster.

Scrum Manager designa por tanto, más que al rol, a la responsabilidad de funcionamiento del modelo. Puede ser a nivel de proyecto o a nivel de la organización; y en algunos casos resultará más apropiado un rol exclusivo (tipo ScrumMaster) y en otros, puede ser mejor que las responsabilidades de funcionamiento las asuman los responsables del departamento de calidad o procesos, o del área de gestión de proyectos, Palacio (2008).

Uno de los roles importantes del Scrum Manager es buscar la mejora continua con scrum ya que de forma regular, se analiza y se determinan acciones para mejorar la configuración de Scrum, Palacio (2008).

De forma continua se monitoriza la implantación y funcionamiento de Scrum en los proyectos y se identifica lo siguiente:

- Impedimentos en los proyectos para que el equipo pueda llevar a cabo el objetivo del sprint.
- Prácticas o decisiones en la organización que impiden o dificultan la metodología Scrum.

Gestión de requisitos con Scrum

Realizando un estudio referente a la gestión de requisitos en otras metodologías como la CMMI; Fernández Y (2009) menciona que de acuerdo a CMMI el propósito del área de procesos de gestión de requisitos consiste en la gestión de los requisitos de los productos y de los componentes del producto del proyecto, e identificar inconsistencias entre esos requisitos y los planes de trabajo del proyecto.

Dentro de esta fase esta la gestión de los cambios en los requisitos que es lo que se pretende estudiar como parte de los objetivos del presente estudio. Según CMMI el propósito de esta gestión es gestionar los cambios a los requisitos a medida que evolucionan durante el proyecto; registrar todos los requisitos y los cambios sobre requisitos que se generan durante el proyecto, manteniendo un histórico de cambios y evaluando el impacto de cambio de los requisitos desde el punto de vista de las partes interesadas, Fernández Y (2009).

Para cumplir con estas descripciones se deben cumplir los siguientes puntos que se nombran a continuación:

- Documentar todos los requisitos y los cambios a los requisitos que son dados o generados por el proyecto.
- Mantener la historia de cambios de requisitos con la razón del cambio.
- Evaluar el impacto de los cambios de requisitos desde el punto de vista de las partes interesadas relevantes.
- Colocar los requisitos y los datos de los cambios disponibles para el proyecto.

Fernández Y (2009) concluye que la gestión de cambios en los requisitos es soportada por Scrum a través de la iteración por sprints. Durante las reuniones de planificación de cada sprint, todos los cambios en los requisitos son actualizados en los backlogs manteniéndose información sobre el cambio, la historia de usuario obsoleta y la nueva historia de usuario. El backlog es accesible por todo el equipo scrum. El cliente está involucrado durante todo el ciclo de vida del producto observando los progresos, aportando ideas, sugerencias o necesidades.

Los productos de trabajo obtenidos de esta práctica son:

- Estado de los requisitos.
- Bases de datos de requisitos: backlogs.
- Bases de datos de decisiones de requisitos: backlogs.

Como resultado de esta fase se pudo estudiar las características más resaltantes de la metodología Scrum, describiendo su ciclo de vida,

profundizando acerca del backlog del producto , los sprints, los roles o responsabilidades de los integrantes del equipo scrum y su manera de trabajar la gestión de requisitos.

En la etapa que se presenta a continuación se describirá el modelo usando ingeniería de métodos en donde se define el modelo producto y el modelo proceso.

Etapa II

Diseñar el modelo de mejora continua para gestión de requisitos usando Ingeniería de métodos

Tomando los principios metodológicos de la ingeniería de métodos definido por Odell (1996) en Sánchez (2011) sabemos que los métodos están compuestos por un modelo producto y por un modelo de procesos como se muestra en la figura 18; a partir de esta teoría se creó el modelo de productos en donde se describen los productos que se desarrollaron y el modelo de procesos que es la descripción de los pasos que se realizan en el método.

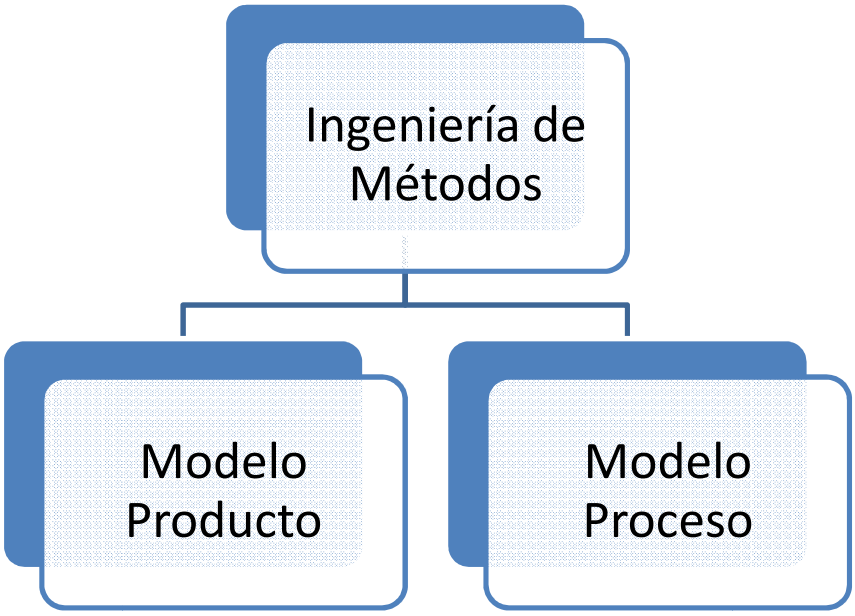


Figura 18. Ingeniería de métodos. Fuente: Sánchez (2011)

Modelo Producto: Es la estructura de los productos que pueden ser entregables, modelos, tablas, diagramas, documentos, etc. que se producen en un método de desarrollo.

Modelo de Procesos: Es la descripción de los pasos que se realizan en el método, pueden ser determinar los requisitos, analizar los requisitos, diseñar el modelo de datos, etc.

Para la realización del modelo de procesos se tomo como referencia el flujo de trabajo del subproceso de gestión de requisitos estudiadas por Montilva (2007) ya que la metodología Scrum no muestra explícitamente estos procesos de manera separada; las mismas están implícitas en varias de las etapas de su ciclo de vida. Montilva (2007), define dentro de la Ingeniería de requisitos el subproceso de gestión de requisitos como se muestra en la figura 19.

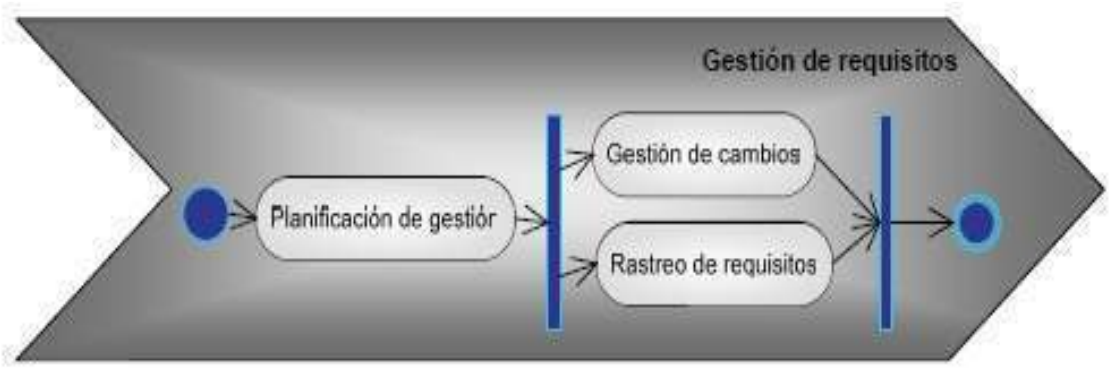


Figura 19. Flujo de trabajo del subproceso gestión de requisitos. Fuente: Montilva (2007).

La planificación de gestión es la planificación y control de actividades de la ingeniería de requisitos, en esta etapa se estiman los costos y se determinan y seleccionan los recursos, por ejemplo las herramientas de apoyo a la ingeniería de requisitos; la otra etapa es la gestión de cambios donde se manejan los cambios en los requisitos una vez que el documento de requisitos ha sido liberado.

En cuanto al rastreo de requisitos Montilva (2007) relaciona los requisitos con el diseño del sistema, sus componentes y las pruebas del sistema y asegura que todos los requisitos son implementados, y afirma que un requisito es rastreable sí se puede determinar quien propuso el requisito, porqué el requisito existe, que otros requisitos se relacionan con él y como el requisito se relaciona con los otros productos del desarrollo del sistema.

Así mismo Montilva (2007) muestra la descripción del flujo de trabajo de gestión de requisitos como se muestra en la tabla 2.

Tabla 2. Descripción del flujo de trabajo de ingeniería de requisitos.

Actividad	Tareas	Productos
Planificación de cambios	1. Definición de los medios de almacenamiento de los requisitos de la aplicación. 2. Establecimiento de procedimientos y mecanismos de mantenimiento y control de requisitos.	1. Procedimientos gestión de requisitos.
Gestión de cambios	1. Seguir los procedimientos y mecanismos para la gestión de cambios de requisitos. 2. Realizar el cambio en los requisitos. 3. Asegurar consistencia e integridad de la base de datos una vez realizados los	1. Base de datos de requisitos actualizada.

	cambios.	
Rastreo de cambios	1. Definir ámbito de influencia del cambio sobre los requisitos de la aplicación. 2. Asegurar actualización de documentos y modelos de la aplicación.	1. Matriz de rastreo de requisitos.

Fuente: Montilva (2007).

La metodología Scrum soporta las distintas actividades mencionadas anteriormente por Montiva (2007) como se puede mostrar en la siguiente tabla 3 de resumen, en donde se mapean las tareas de gestión de requisitos definidas por Montilva (2007) y las actividades estudiadas y analizadas de la metodología Scrum por el autor.

Tabla 3. Aplicando Scrum a la gestión de requisitos.

Actividad	**Tareas Gestión requisitos**	**Aplicando Scrum**	**Productos Scrum obtenidos**
Planificación de Cambios	1. Definición de los medios de almacenamiento de los requisitos de la aplicación. 2. Establecimiento de procedimientos y mecanismos de mantenimiento y control de requisitos.	1. En Scrum ya se tiene definido que el almacenamiento es en los Backlog.	1. Product Backlog.
Gestión de cambios.	1. Seguir los procedimientos y mecanismos para la gestión de cambios de requisitos. 2. Realizar el cambio en los requisitos. 3. Asegurar consistencia e integridad de la base de datos una vez realizados los cambios.	1. Iteración por Sprints. 2. Cambios en los requisitos actualizados en el backlog del producto.	1. Estado de los requisitos. 2. Base de datos de los requisitos (Backlog). 3. Base de datos de decisiones de los requisitos.
Rastreo de Cambios	1. Definir ámbito de influencia del cambio sobre los requisitos de la aplicación. 2. Asegurar	1. El ciclo de vida scrum maneja la trazabilidad entre los sprints y los	1. La Matriz de trazabilidad de los requisito

	actualización de documentos y modelos de la aplicación.	entregables. 2. A través de los backlogs de producto y de los sprints es posible trazabilidad bidireccional entre las historias de usuario y los entregables. 3. También es posible realizar un seguimiento de las historias de usuario completadas en un determinado sprint	s en scrum viene siendo suplida por backlogs e historias de usuario. 2. Por medio de los los Backlogs realizan el seguimiento de los requisitos.

Fuente: El Autor

Para el modelo de procesos se han tomado las etapas de gestión de requisitos propuestas por Montilva (2007) como se muestra en la figura 20 donde puede observarse el modelo de procesos.

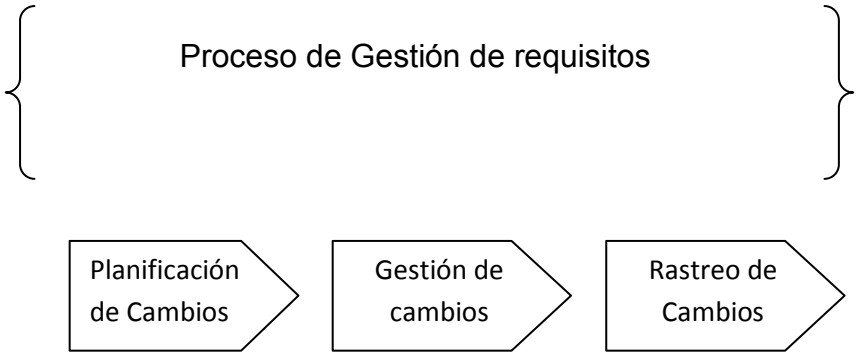


Figura 20. Modelo de Procesos Gestión de requisitos. Fuente: Montilva (2007)

A este modelo de procesos se le agrego un proceso que usa la metodología Scrum para la mejora continua del proceso llamado retrospección y que el autor ha llamado mejora continua usando retrospección quedando el modelo de procesos de la siguiente como se muestra en la figura 21.

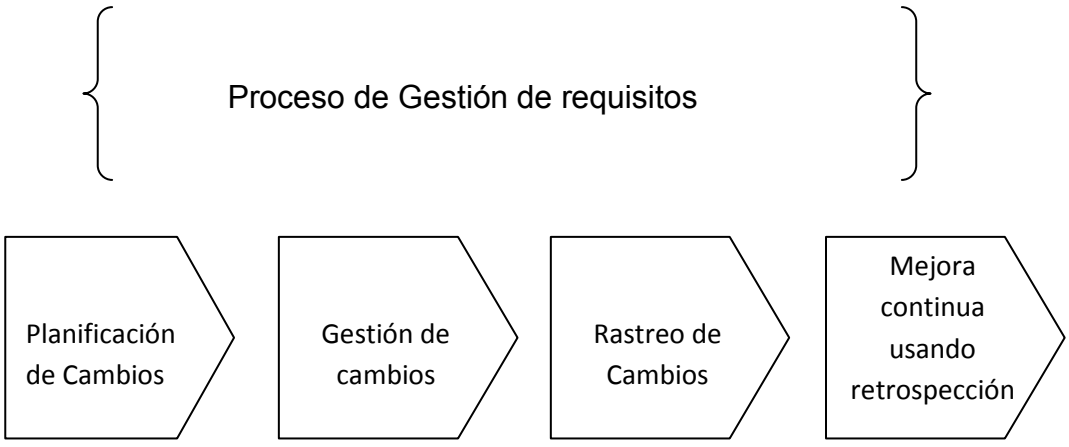


Figura 21. Modelo de procesos propuesto. Fuente: El autor

A continuación se van a describir cada una de estas actividades del modelo de procesos propuesto resolviendo sus tareas de gestión de requisitos usando Scrum y se mostrara el diagrama de actividades de cada uno de estos procesos.

El modelo de gestión de requisitos va a cumplir con las siguientes actividades que se muestran a continuación en la tabla 4:

Tabla 4. Actividades por fase del modelo.

Actividades	**Gestión de Requisitos**
Actividad 1	Planificación de Cambios
Actividad 2	Gestión de cambios
Actividad 3	Rastreo de Cambios
Actividad 4	Mejora continua usando retrospección

Fuente: El autor

Actividades de la fase de gestión de requisitos

Actividad 1 – Fase Planificación de Cambios.

Esta actividad tiene como objetivo la planificación y control de las actividades de la ingeniería de requisitos como lo son:

- Elaborar el plan de Ingeniería de requisitos.

- Estimar costos.
- Se Determinan y seleccionan los recursos por ejemplo las herramientas de apoyo a la ingeniería de requisitos.

La metodología Scrum realiza una planificación de cambios de manera ágil en dos de sus etapas resolviendo lo que propone Montilva (2007) para esta fase ; en la etapa de concepto de Scrum se decide y se seleccionan las personas que llevaran a cabo la realización del producto y en la etapa de especulación también de scrum de determinan las limitaciones impuestas por el entorno de negocio (costos y agendas), en la figura 22 se muestra el diagrama de actividades de la fase de planificación de cambios.

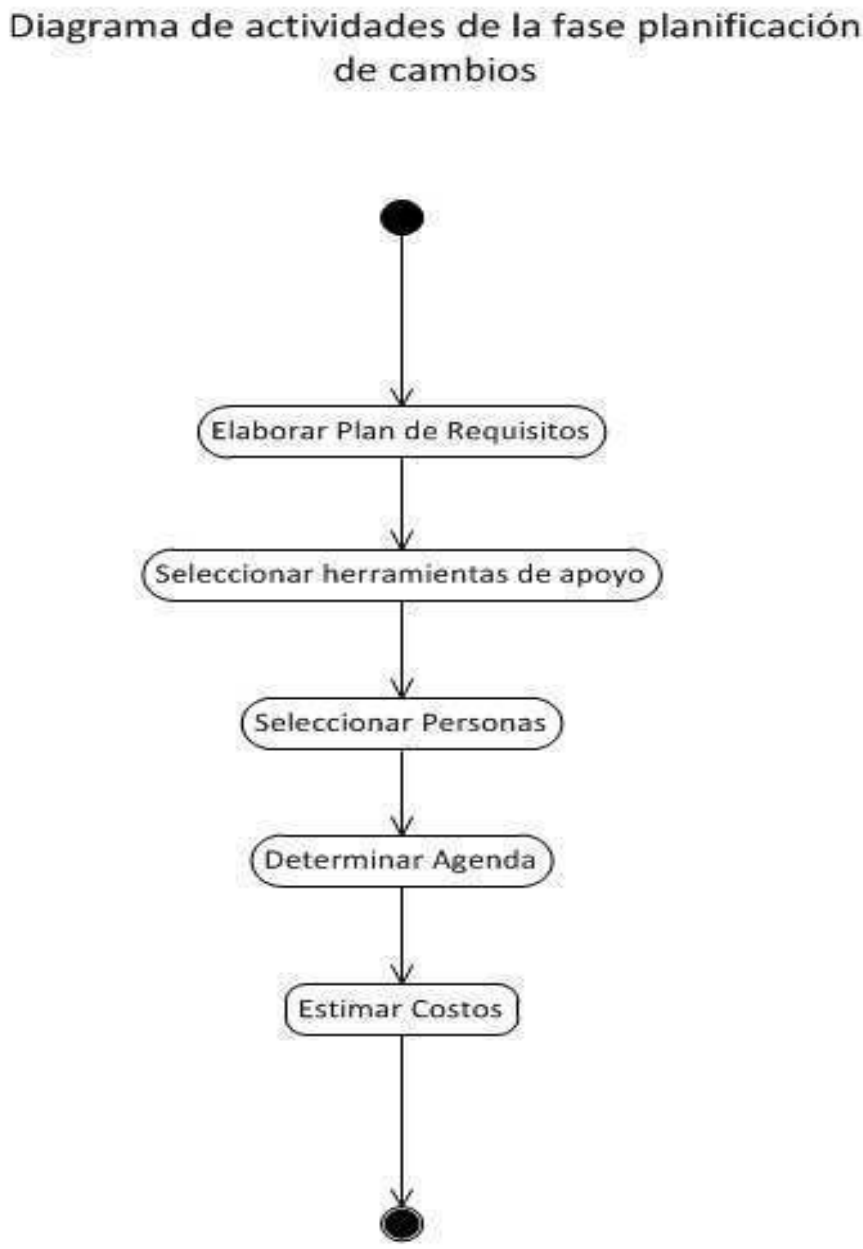


Figura 22. Diagrama de actividades de la fase planificación de cambios.

Fuente: El autor.

En la tabla 5 se observan las entradas, salidas y responsables de la actividad Fase Planificación de cambios.

Tabla 5. Resumen Actividad Planificación de cambios.

Entradas	1. Documento de captura de requisitos, historias de usuario.
Salidas	1. Documento de plan de requisitos. . 2. Documento de versiones y fecha de entrega. 3. Documento de costos.
Responsables	1. Propietario del producto (Product Owner). 2. Equipo de desarrollo. 3. Scrum Master.

Fuente: El autor

Actividad 2 – Fase de Gestión de cambios

En esta actividad se siguen los procedimientos y mecanismos para la gestión de cambios de requisitos, se realiza el cambio de los requisitos y se asegura la consistencia e integridad de la base de datos una vez realizados los cambios

Esta actividad es soportada por Scrum a través de la iteración por sprints en donde se realiza las reuniones por cada sprint, en estas reuniones de planificación de cada sprint los cambios en los requisitos son actualizados en

los backlogs guardando la información sobre el cambio y manteniéndose la historia obsoleta y la nueva historia de usuario. En la figura 23 puede observarse el diagrama de actividades de la gestión de cambios.

Diagrama de actividades de la fase de gestión de cambios.

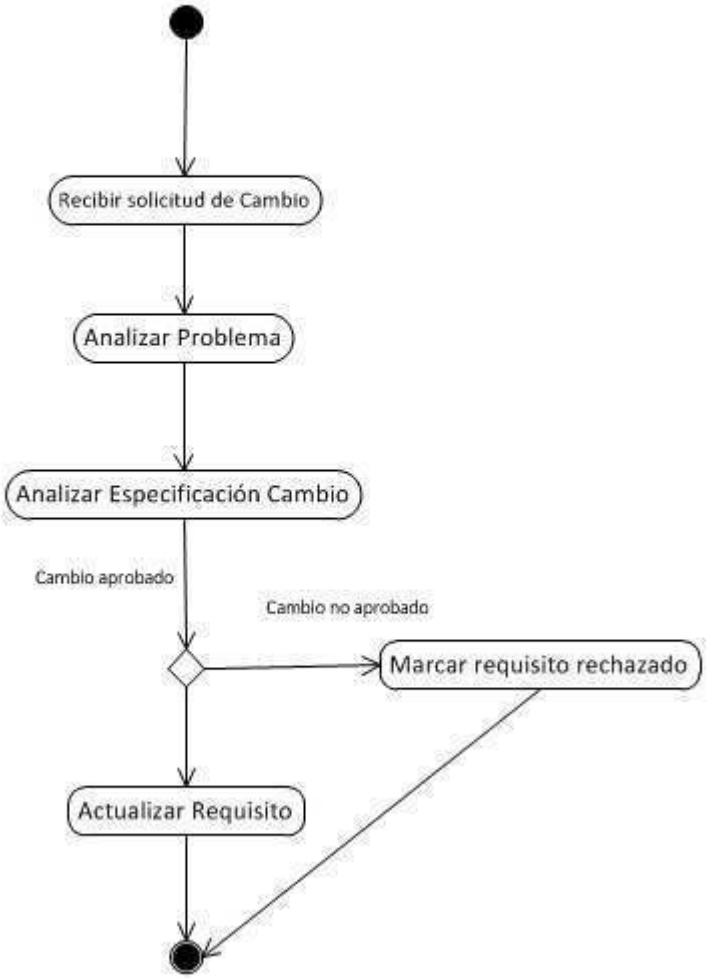


Figura 23. Fase de Gestión de cambios. Fuente: El autor.

En la tabla 6 se muestran las entradas, salidas y responsables de la fase de gestión de cambios.

Tabla 6. Resumen actividad gestión de cambios.

Entradas	1. Sprint Backlog
Salidas	1. Nuevas Historias de usuario a implementar 2. Backlog del sprint 3. Estado de los requisitos. 4. Bases de datos de los requisitos (Backlogs). 5. Bases de datos de las decisiones de los requisitos (Backlogs).
Responsables	1. Equipo de desarrollo. 2. Scrum Master. 3. Propietario del producto.

Fuente: El autor

Actividad 3 - Fase de rastreo de cambios.

En esta actividad se Define el ámbito de influencia del cambio sobre los requisitos de la aplicación y se asegura la actualización de los documentos y modelos de la aplicación.

Según lo estudiado en la etapa anterior el rastreo o trazabilidad es soportada por Scrum de la siguiente manera:

- El ciclo de vida scrum maneja la trazabilidad entre los sprints y los entregables.
- A través de los backlogs de producto y de los sprints es posible trazabilidad bidireccional entre las historias de usuario y los entregables.
- También es posible realizar un seguimiento de las historias de usuario completadas en un determinado sprint.

En la figura 24 se puede observar el modelo de procesos de fase de rastreo de cambios.

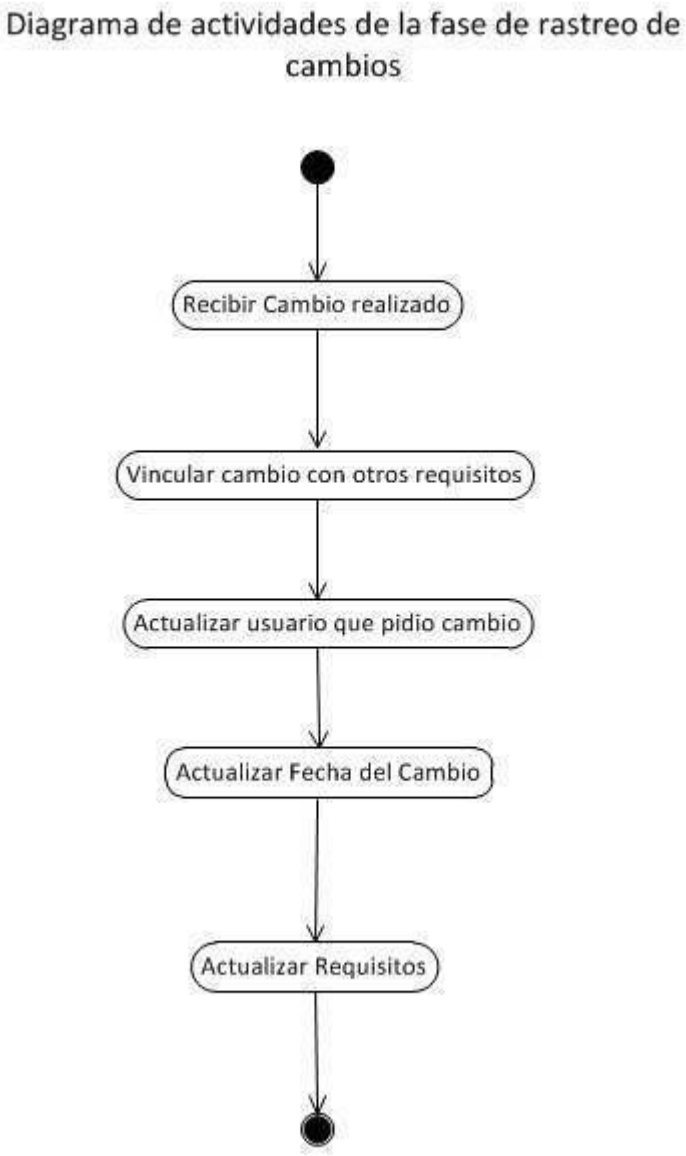


Figura 24. Fase de rastreo de cambios. Fuente: El autor

En la tabla 7 se muestran las entradas, salidas y responsables de la actividad de rastreo de cambios.

Tabla 7. Cuadro resumen actividad de rastreo de cambios.

Entradas	1. Sprint Backlog 2. Documento avances del Sprint.
Salidas	1. Documento de Avances, 2. Nuevas Historias de Usuario.
Responsables	1. Equipo de desarrollo, 2. Scrum Master.

Fuente: El autor

Actividad 4 – Fase de mejora continua usando retrospección.

Esta actividad tiene como objetivo, mejorar de forma continua el proceso de desarrollo, se analizan los objetivos propuestos inicialmente en el backlog del sprint revisando los aspectos positivos y negativos que sirven de retroalimentación para aplicar cambios y ajustes necesarios en el siguiente sprint del proceso.

En la figura 25 puede observarse el modelo de actividades del proceso de retrospección.

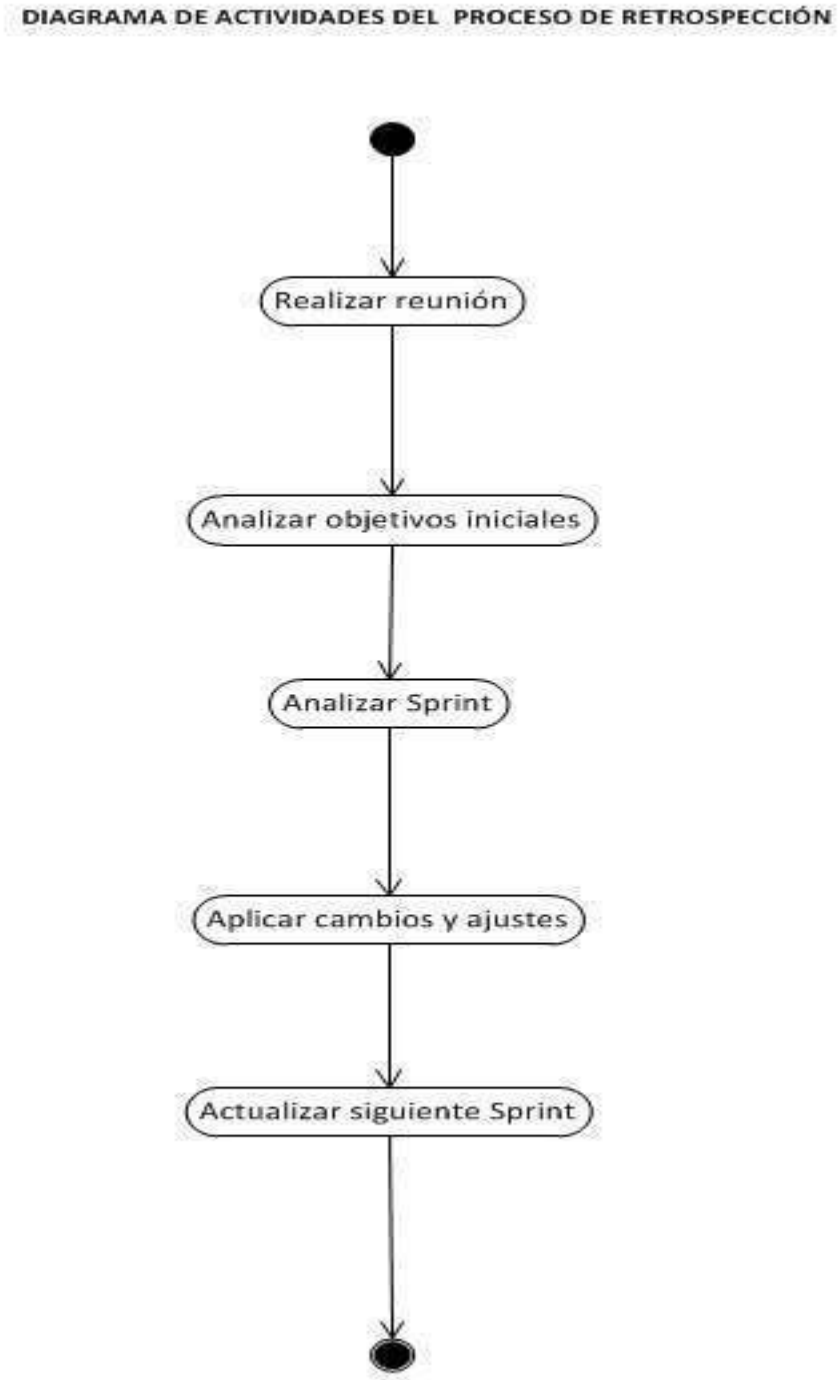


Figura 25. Actividades del proceso de retrospección. Fuente**:** El autor.

En la tabla 8 se muestran las entradas, salidas y responsables de la actividad del proceso de retrospección.

Tabla 8. Resumen de Actividad retrospección.

Entradas	1. Sprint Backlog.
Salidas	1. Documento de aspectos positivos y negativos del sprint. 2. Documento de cambios y ajustes con mejoras para el próximo sprint. .
Responsables	1. Equipo de desarrollo. 2. Scrum Master.

Fuente: El autor.

En la figura 26 se muestra como sería el modelo producto del modelo propuesto, donde este se divide en los productos de trabajo que están compuestos por las historias de usuario, sprint backlog y product backlog y en los productos finales que serían los entregables entre los cuales estarían nuevas historias de usuario, requisitos clasificados, estado de los requisitos, bases de datos de los requisitos y modelado en herramienta BPMS.

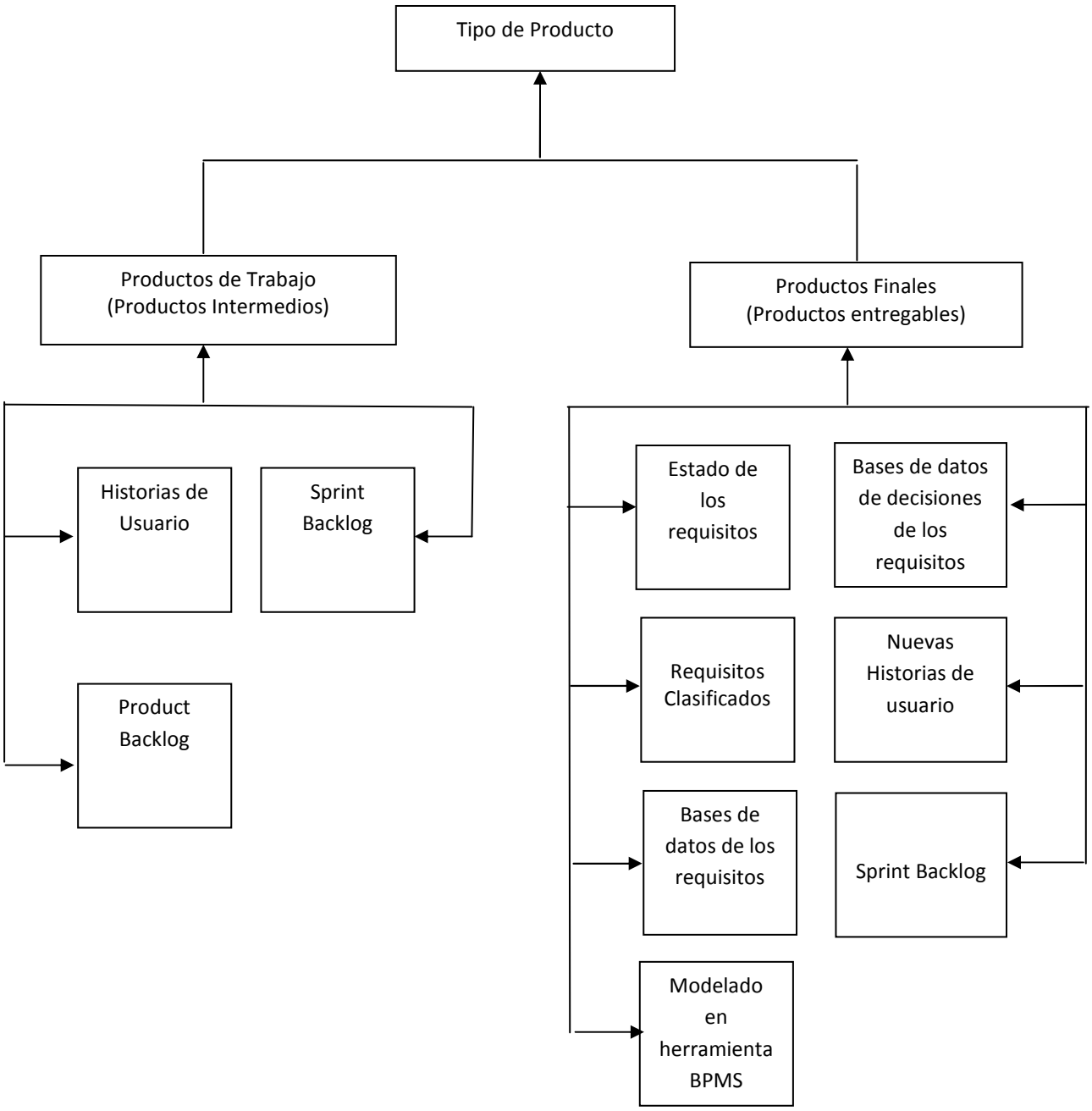


Figura 26. Modelo de producto del modelo propuesto. Fuente: El autor.

Este modelo de productos genera un conjunto de productos de trabajo o productos intermedios entre los que tenemos las historias de usuario, el product backlog y el sprint backlog, así mismo genera unos productos finales

como el estado de los requisitos, clasificación de requisitos, bases de datos de los requisitos, bases de datos de las decisiones de los requisitos y nuevas historias de usuario.

Historias de usuario

Las historias de usuarios representan las funcionalidades definidas entre los clientes y los miembros del equipo Scrum, son usadas en varias de las etapas del ciclo de vida de Scrum, las historias de usuario son priorizadas y pasan a formar parte del Backlog del producto, son agrupadas en sprints según su prioridad.

Product Backlog

En el product backlog se encuentran todas las funcionalidades, mejoras y corrección de errores que debe realizar el equipo del proyecto para mejorar el producto.

Sprint Backlog

El sprint backlog es usado para descomcoponer las funcionalidades del product backlog para realizar una parte completa y operativa del producto., aquí se asigna cada tarea a una persona y se indica cuanto tiempo de trabajo se necesita para culminar la tarea.

Estado de los requisitos

Con Scrum es posible saber el estado de los requisitos ya que permite realizar un seguimiento de las historias de usuario completadas en un determinado sprint por lo que se puede saber cuáles historias de usuario están completadas y cuáles no lo están.

Clasificación de los requisitos

La clasificación de los requisitos es realizada por Scrum en su fase de planificación donde se realiza una reunión por cada sprint donde se establecen las historias de usuario a ser realizadas, participando de esta reunión los clientes, jefe de proyecto y desarrolladores, en dicha etapa se mueven las historias de usuario con mayor prioridad para el cliente al backlog del sprint para que entren en la fase de desarrollo.

Bases de datos de los requisitos

Scrum usa para almacenar los requisitos el product backlog, donde se encuentran todos los requisitos almacenados, sería conveniente contar con una herramienta de gestión de requisitos para el almacenamiento de los mismos.

Como resultado de esta fase se obtuvo la definición y modelado de los procesos asociados al ciclo de vida con que trabaja la metodología Scrum aplicados a la gestión de requisitos usando UML para la realización de los respectivos diagramas de actividades de cada uno de los procesos del modelo de procesos obtenido en la etapa anterior y obteniendo tablas de resumen con las entradas, salidas y responsables de cada uno de los procesos.

En la siguiente etapa de la investigación se llevaran estos procesos a una herramienta BPMS, donde se realizara el modelado de cada proceso usando la notación de BPMN2 buscando facilitar el seguimiento y monitoreo de procesos de la gestión de requisitos.

Etapa III

Modelado del modelo propuesto bajo un enfoque BPM

En esta etapa los procesos de negocio diseñados en la fase anterior se modelan como flujos de trabajo con la notación BPMN2. Estos procesos fueron modelados usando la herramienta BPMS BonitaSoft la cual es una solución abierta que proporciona características para diseñar, desarrollar, ejecutar y supervisar procesos de negocio.

En primer lugar se muestra el modelado del proceso de la planificación de cambios, como se muestra en la figura 27. Este proceso ejecuta las actividades asociadas a la planificación de .cambios.

Este proceso permite construir un plan para la gestión de requisitos, seleccionando las herramientas de apoyo, escogiendo los desarrolladores a ser asignados para el desarrollo de las tareas del sprint backlog, fijar los tiempos para la realización de cada una de las tareas a realizar y estimar los costos necesarios para la realización del producto.

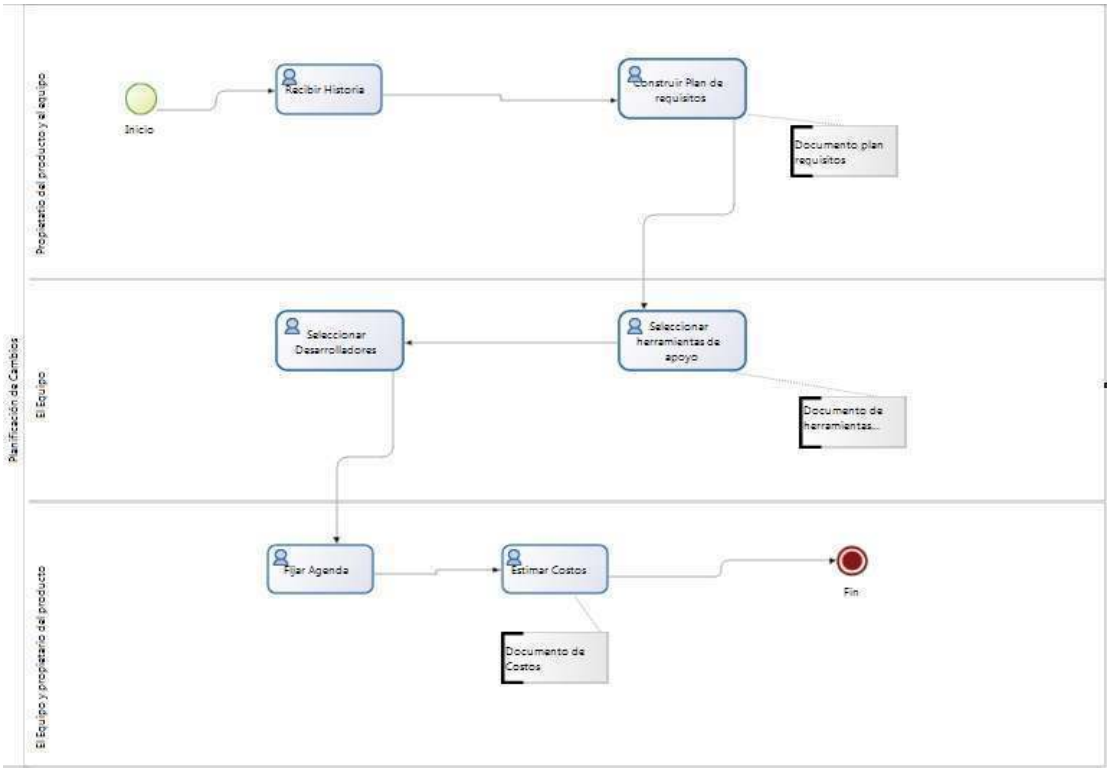


Figura 27. Modelado del Proceso de Planificar Cambios. Fuente: El autor.

Seguidamente, en la figura 28 se muestra las actividades del proceso de gestión de cambios con la notación BPMN2. Este proceso permite analizar el cambio al requisito que se quiere realizar, decidir si es posible realizar el cambio o no es posible, aprobar el cambio y actualiza el requisito con los cambios.

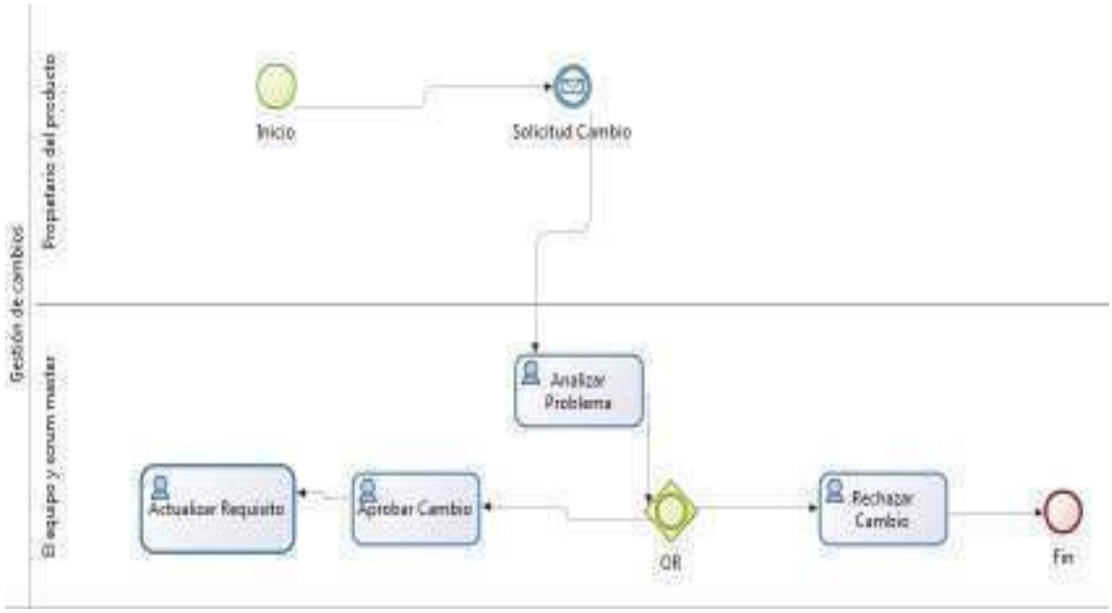


Figura 28. Modelado del proceso gestión de cambios. Fuente: El autor

Posteriormente en la figura 29 se muestran las actividades de rastreo de cambios de los requisitos, en donde se recibe el cambio que se quiere realizar en el requisito, se vincula el cambio con los requisitos relacionados, se actualiza el usuario que pidió el cambio y la fecha del cambio y por último se actualizan los requisitos.

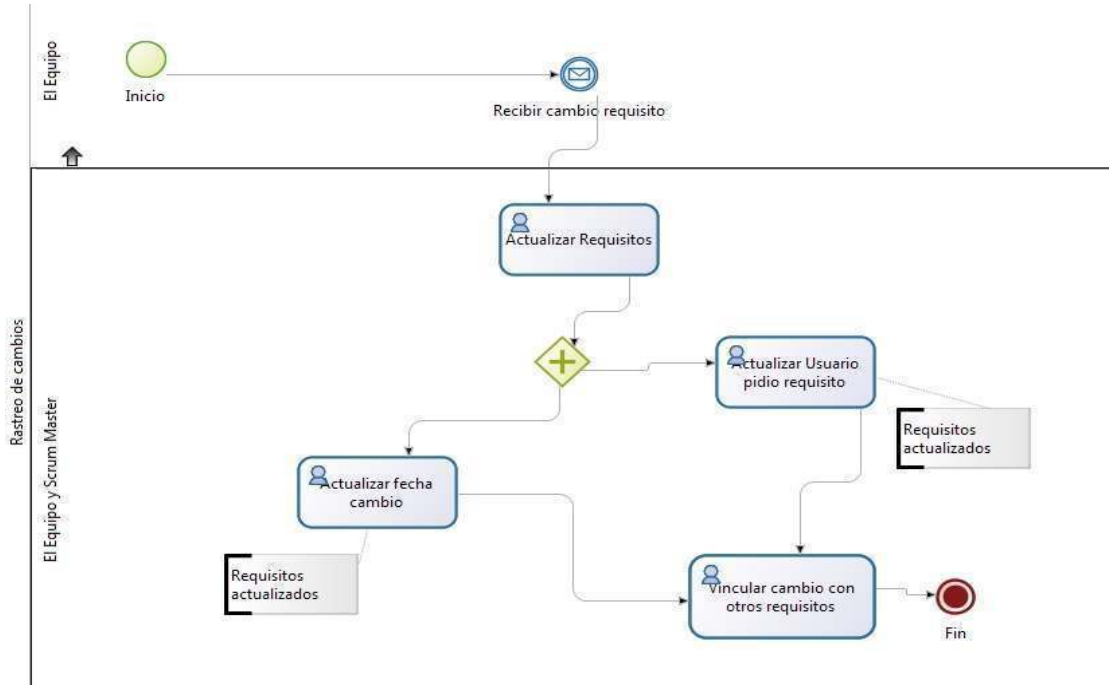


Figura 29. Modelado del proceso rastreo de cambios. Fuente: El autor.

Por último en la figura 30 se muestran el proceso de mejora continua usando retrospección, este proceso permite mejorar de forma continua el proceso de desarrollo, se analizan los objetivos propuestos en el backlog del sprint de manera de obtener los aspectos positivos y negativos que sirven de retroalimentación para aplicar cambios y ajustes necesarios en el siguiente sprint del proceso.

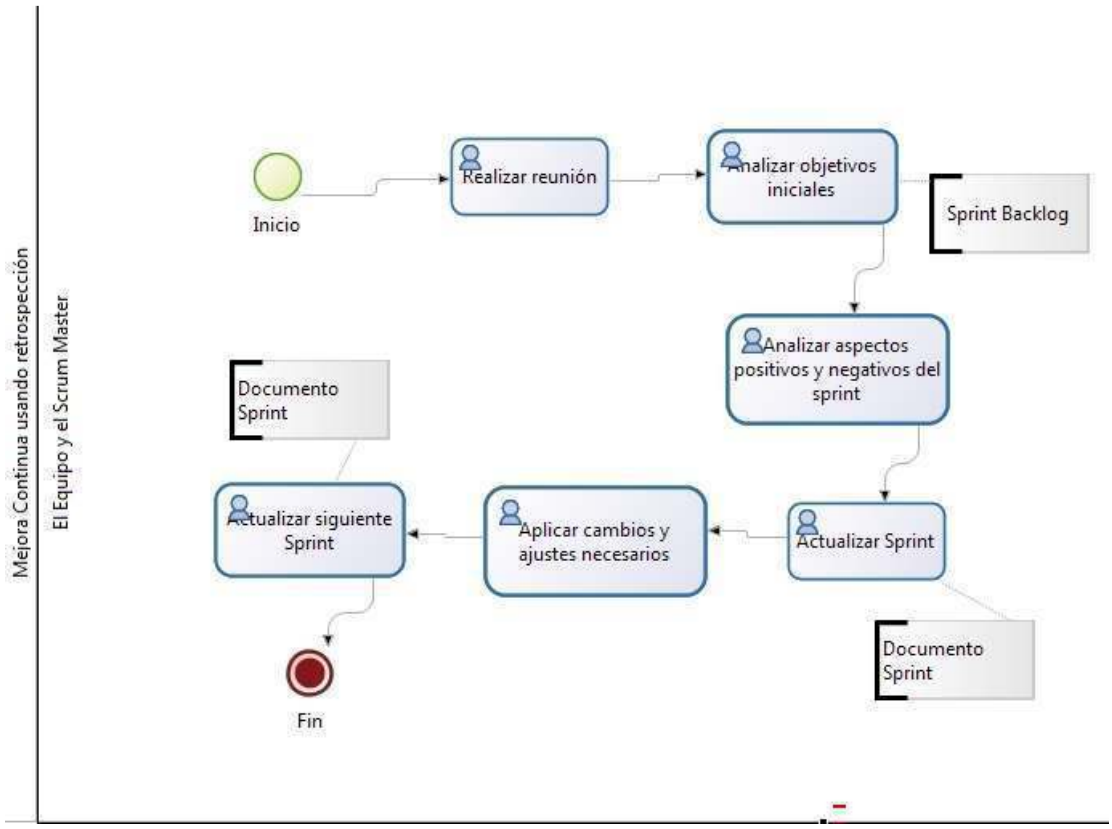


Figura 30. Modelado mejora continua usando retrospección. Fuente: El autor.

En la última fase del proyecto se presentan los indicadores diseñados para medir el proceso.

Etapa IV

Diseñar indicadores de desempeño del modelo propuesto.

Los indicadores de desempeño del modelo de gestión de requisitos van a permitir a través de ellos colaborar con la mejora continua del modelo, en esta etapa se han diseñado una seria de indicadores que van a contribuir con esto.

Vogel (2007) en Valdivia M, establece una ruta metodológica para la creación de indicadores que se basa en cuatro pasos fundamentales; la determinación del objetivo del indicador donde se declaran los elementos críticos para el éxito de la estrategia, luego se procede a aclarar de forma concisa cuál es el objetivo buscado con la definición del indicador, el tercer paso es buscar las variables críticas del objetivo, y por último hallar los indicadores adecuados para cada variable. En la Tabla 9 puede observarse esta ruta metodológica para establecer indicadores.

Tabla 9. Ruta metodológica para establecer indicadores.

Objetivo	Declaración de lo que la estrategia debe lograr y qué es crítico para su éxito.
Aclarar	Qué queremos realmente conseguir (aclarar cuál es el objetivo buscado).

Variables que muestren logros	Hallar las variables críticas del objetivo buscado (Cómo nos damos cuenta que lo estamos buscando).
Indicador	Hallar los indicadores adecuados para cada variable. ¿Cuáles son los indicadores críticos que indican nuestra dirección estratégica?

Fuente: Vogel (2007) en Valdivia.

A partir de la ruta metodológica mostrada en la tabla 9 se diseñaron los indicadores de desempeño del modelo de mejora continua para la de gestión de requisitos, el primer indicador se muestra en la tabla 10.

Tabla 10. Indicador de mejora continua.

Objetivo	Analizar los objetivos del backlog del sprint.
Aclarar	Se busca los aspectos positivos y negativos del sprint
Variables que muestren logros	Si se están realizando mejoras en el siguiente sprint con los aspectos negativos encontrados.
Indicador	# de aspectos positivos y negativos encontrados. # de mejoras realizadas.

Fuente: El autor.

Este indicador tiene las siguientes características como se muestran en la tabla 11.

Tabla 11. Características del indicador de mejora continua

Uso del indicador	Medir la cantidad de mejoras realizadas al proceso.
Valores esperados	Por cada aspecto negativo encontrado debería existir una mejora a realizar en el siguiente sprint.
Cómo regula el proceso	Si existe algún aspecto negativo al que no se le haya encontrado una mejora se debe trabajar en encontrar una mejora lo más rápido posible.
Alertas	Si por cada 2 fallas encontradas no existe al menos una mejora a realizar.

Fuente: El autor.

El segundo indicador diseñado se muestra en la tabla 12.

Tabla 12. Indicador de productos entregados.

Objetivo	Entregar un producto probado y funcionado.

Aclarar	Un modelo de gestión de requisitos que ayude al proceso de desarrollo a entregar el producto cada 4 semanas o menos.
Variables que muestren logros	Si el cliente está satisfecho con las fechas de entrega y las funcionalidades realizadas.
Indicador	# de productos entregados. # de clientes satisfechos.

Fuente: El autor.

Este indicador tiene las siguientes características como se muestra en la tabla 13.

Tabla 13. Características del Indicador de productos entregados.

Uso del indicador	Medir la satisfacción del cliente en relación a los productos que le son entregados, así como también medir la cantidad de productos realizados por el equipo de trabajo.
Valores esperados	Mayor o igual al 90% de clientes satisfechos.
Cómo regula el proceso	Si el porcentaje de clientes satisfechos es menor del 90 % se debe realizar un estudio del porque los clientes no están quedando satisfechos con el producto entregado y aplicar

	las correcciones necesarias.
Alertas	Menos del 90% de clientes satisfechos.

Fuente: El autor

A continuación se muestra el tercer indicador diseñado en la tabla 14.

Tabla 14. Indicador de Backlog del producto.

Objetivo	El propietario del producto tiene un backlog del producto.
Aclarar	Un modelo de gestión de requisitos donde el rol del propietario del producto de Scrum cuente con el backlog del producto .
Variables que muestren logros	Si el cliente está satisfecho con las prioridades que se les da a sus historias de usuario más relevantes para él. Si el quipo realiza las estimaciones en los sprint.
Indicador	# de clientes satisfechos con las prioridades que se les da a sus historias de usuario .

Fuente: El autor.

En la tabla 15 se muestra las características del indicador de Backlog del producto.

Tabla 15. Características del indicador Backlog del producto.

Uso del indicador	Medir el nivel de satisfacción de los clientes de acuerdo con las prioridades en sus pedidos
Valores esperados	Mayor al 80 % de clientes satisfechos.
Cómo regula el proceso	Si el porcentaje de clientes satisfechos es menor del 80 % se debe revisar las prioridades que se le están asignando a las tareas en desarrollo y ajustar las mismas.
Alertas	Menos del 80% de clientes satisfechos.

Fuente: El autor

Esta fase permitió diseñar los indicadores de desempeño para el modelo de gestión de requisitos que ayuden a tener métricas que permitan la mejora continua de sus procesos.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

La ingeniería de requisitos es una de las etapas fundamentales dentro del proceso de desarrollo de software para cumplir con éxito las metas y necesidades que los usuarios solicitan en el día a día para sus sistemas de información, para ello se hace necesario la creación de métodos y modelos que permitan cubrir estas necesidades de una manera ágil en este mundo cambiante que enfrentamos hoy en día; para ello es fundamental el uso de modelos de requisitos de software que usen la mejora continua de manera de tomar los aspectos positivos obtenidos en cada iteración del modelo y conociendo los aspectos negativos de manera de ir mejorando en cada iteración del proceso.

Scrum es una metodología ágil que ofrece mejora continua en sus ciclos de vida siendo una opción interesante para las empresas que hoy en día desarrollan software y que no quieren invertir tanto tiempo y esfuerzo en metodologías más pesadas. Esta investigación permitió unir gestión de requisitos con Scrum, partiendo de las actividades de gestión de requisitos definidas por Montilva (2007) que proponen que se debe hacer en la gestión de requisitos y mapeándola contra Scrum que plantea una forma de definir como se pueden resolver las actividades que se requieren en la gestión de requisitos.

Por lo anteriormente planteado fue posible obtener las siguientes conclusiones:

- La metodología Scrum no por ser una metodología ágil le da poca importancia a los requisitos, sin embargo parte del hecho de que los requisitos van a estar cambiando a lo largo de todo el ciclo de vida del desarrollo de software.

- Los requisitos en Scrum son usados en cada una de sus etapas y en cada una de sus actividades de su ciclo de vida.

- El uso de las metodologías de software para nuestros procesos desarrollo de software es un elemento importante para conseguir la satisfacción de los clientes que requieren el producto de software.

- El uso del modelado en BPMN permite a los responsables de los procesos conocer de una manera clara y estandarizada con una notación sencilla los procesos de negocio que se llevan en la empresa y permiten monitorizarlos, controlarlos y automatizarlos.

Por otro lado, se determinaron una serie de recomendaciones que pueden servir para futuras investigaciones o para continuar dándole profundidad al presente trabajo, las cuales son mencionadas a continuación:

- Esta investigación se puede continuar realizando el modelo de gestión de requisitos usando otras metodologías ágiles bien sea Xp, FDD, Crystal y otros.

- Completar la investigación para otras etapas de la ingeniería de requisitos como la captura y la validación.

- Implantar las actividades y procesos definidos en esta investigación en un entorno de desarrollo real, para validar los procesos aquí planteados.

- Se puede automatizar el proceso usando la herramienta BPMS para poder realizar un seguimiento más controlado en cada una de las actividades definidas para los procesos del modelo.

- En futuros trabajos de investigación se sugiere agregar nuevos indicadores que complementen los ya existentes.

REFERENCIAS BIBLIOGRAFICAS

Acosta, C. (2010). Diseño e implementación de una herramienta de representación del conocimiento para apoyar la gestión de requisitos en un proceso de desarrollo de software. Tesis de Magister en Ciencias mención Computación, Universidad de Chile.

Andriano, N. (2006). Comparación del proceso de elicitación de requerimientos en el desarrollo de software a medida y empaquetado. Propuesta de métricas para la elicitación. Tesis de Magister en Ingeniería de software en facultad de informática, Universidad Nacional de la Plata.

Balestrini, M. (2006). Como se elabora el proyecto de investigación**.** Caracas**:** Consultores Asociados.

Bazán, P. (2009). Un modelo de integrabilidad con SOA y BPM**.** Tesis de Maestría en redes de datos, Universidad Nacional de la Plata.

Bennet, S, y Otros (2007). Análisis y diseño orientado a objetos de sistemas usando UML. McGraw-Hill.

Bernal, R (2007). Ingeniería de requisitos en los métodos de desarrollo ágiles.Universidad de Sevilla, España.

Berrocal, J. y Otros (2010). Usando técnicas BPM para agilizar la gestión de procesos de Software y mejorar la alineación con el negocio. Universidad de Extremadura. Acta de los talleres de las jornadas de Ingeniería de Software y Bases de Datos, Vol.4, No. 4, 2010.

Berrocal, J. y Otros (2009). Patrones para la extracción de Casos de Uso a partir de Procesos de Negocio. Universidad de Extremadura. Acta de los talleres de las jornadas de Ingeniería de Software y Bases de Datos, Vol.3, No. 3, 2019.

Berrocal, J. y Otros. Hacia una gestión del proceso de software dirigida por procesos de negocio. Universidad de Extremadura. Dpto de Ingeniería de sistemas informáticos y telemáticos.

Borland (2005). MITIGATING RISK WITH EFFECTIVE REQUIREMENTS ENGINEERING How to improve decision-making and opportunity through effective requirements engineering. Part two in a series about understanding and managing risk. April 2005.

Brinkkemper, S. (1996). *Method engineering: engineering of information systems development methods and tools*. University of Twente . Information and Software Technology. [Article en lineal] Disponible: http://doc.utwente.nl/18012/1/Brinkkemper96method.pdf

Canós, J. y Otros (S.F). Metodologías ágiles en desarrollo de Software. Universidad Politécnica de Valencia.

Choque, G (2001). Ingeniería de Requerimientos Disponible en <http://www.umsanet.edu.bo/docentes/gho-que/Art_IngRequerim.pdf>>

Evans, E. (2004). Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison Wesley.

Fernández, Y. (2009). Estudio sobre la correspondencia entre prácticas CMMI y prácticas ágiles y su aplicación en pymes, Tesis de Máster en tecnologías de la información, Universidad Politécnica de Madrid.

Garimella, K. y otros (2008). Introducción a BPM**.** Indianápolis, Indiana: Wiley Publishing, Inc.

Gonzalez, O (2011). Un acercamiento a la trazabilidad en el desarrollo ágil de software. Revista Cubana de ciencias informáticas (RCCI).

Harmsen, A. (1997). *Situational Method Engineering*. University of Twente. Netherlands.

Inostroza, V (2010). Programación de pabellones quirúrgicos en hospitales públicos con tecnologías de ejecución de modelos de procesos. Proyecto de grado para optar al grado de Magister en Ingeniería de Negocios con tecnologías de información, Universidad de Chile.

Leffingwell, Dean (2011). Agile Software Requirements. Indianápolis, Indiana: Addison Wesley.

Mcdonald Landazuri, Bárbara A. 2005. Definición de Perfiles en Herramientas de Gestión de Requisitos. Facultad de Informática Universidad Politécnica de Madrid. Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software.

Merchan, L. y otros (2008). Definición de una metodología ágil de Ingeniería de requerimientos para empresas emergentes de desarrollo de software del sur occidente colombiano. Revista científica Guillermo de Ockham. Vol 6, No 1.

Montilva, J (2007). Taller de Ingeniería de Requisitos. Universidad de los Andes.

Montilva, J (2007). Desarrollo de softtware empresarial. Universidad de los Andes.

Odell, J. (1996). A primer to method Ingenieering. INFOSYS. USA.

Palacio, J. (2008). Flexibilidad con Scrum. Adaptando los procesos a la empresa.

Pelayo, Cristina (2007). TALISMAN: Desarrollo Ágil de Software con Arquitecturas Dirigidas por Modelos. Tesis Doctoral, Universidad de Oviedo.

Peralta, A (2003). Metodología Scrum. Universidad ORT Uruguay. Facultad de Ingeniería.

Piña, Sara (2011). Metodología para la gestión de requisitos basada en el modelo CMMI en una organización de software. Caso grupo corporativo Marna. Proyecto de grado para optar al título de Magister en Ciencias de La computación, Universidad Centroccidental Lisandro Alvarado, Barquisimeto.

Rodríguez, P. (2008). Estudio de la aplicación de metodologías ágiles para la evolución de productos de software. Tesis de Máster en tecnologías de la información, Universidad Politécnica de Madrid.

Rolland, C. (1997). *A primer for method engineering*. Conference INFORSID. Universite de Paris. Toulouse.

Ros, J. (2009). Una propuesta de gestión integrada de modelos y requisitos en líneas de productos de software. Tesis Doctoral, Universidad de Murcia.

Sanchez, I. (2011). Extensión del método Gray Watch basado en el estádar de calidad ISO/IEC 25010. Proyecto de grado para optar al título de Magister en Ciencias de la computación. Universidad Centrooccidental Lisandro Alvarado, Barquisimeto.

Silva, R. 2005. BPMS La Nueva Plataforma de Misión Crítica. URL: http://www.delfos.co.cu (Consulta: Marzo 09, 2010)

Sivira, B. (2011). Diseño de un plan de mejora continua en un proceso de desarrollo de software basado en Rup y usando prácticas BPM y CMMI. Proyecto de grado para optar al título de Magister en Ciencias de la computación. Universidad Centrooccidental Lisandro Alvarado, Barquisimeto.

Sommerville, I. (2005). Ingeniería del Software**.** Madrid: Pearson Educación, S.A.

Standish Group, R (2009). Standish Group Report CHAOS.

Stephen A. White, "Introduction to BPMN". IBM Corporation. http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf . 2004 (al 16/10/2009)

Universidad Pedagógica Experimental Libertador. (2006) Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales. Cuarta Edición: Barquisimeto, Venezuela.

Valdivia, M y otros(s.f). Fundamentación para la creación de indicadores de selección para herramientas BPM/SOA de software libre.Universidad de las ciencias informáticas, La Habana.

Valverde, A y Sánchez, A (2007). Gestión por procesos (BPM) usando mejora contínua y reingeniería de procesos de negocio. Tesis para optar el título de Ingeniero en Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima.

Vázquez, González y León (s.f.). Proceso de implementación de una plataforma BPM (Bussiness Process Management). Politecnico Colombiano, Medellin.

Villena, A (2008). Un modelo empírico de enseñanza de las metodologías ágiles. Tesis para optar al grado de magister en ciencias de la computación. Universidad de Chile.

Wiegers K. E. (1999). Software Requirements. Microsoft Press.

Zolghadar, M. 2004. Business Process Management and the Need for Measurements. Universidad Lund. Suiza.

ANEXOS

ANEXO A. CURRÍCULO VITAE DEL AUTOR

Resumen Curricular

Datos Personales

Apellidos: Iribarren Páez
Nombres: Oscar Eduardo
Cédula de Identidad: V – 13.189.909
Fecha de Nacimiento: 8 de Noviembre de 1.976
Edo. Civil: Casado
Nacionalidad: Venezolano
Teléfono: (0251) 2636044 – (0414) 5297238- (0241) 8423666
Correo Electrónico: ioscar25@hotmail.com, ioscar76@gmail.com.
Dirección: Av Nectario María, Urb Roca del Valle III, Calle 11
Casa 17 - Cabudare – Edo. Lara

Estudios Realizados

Año 2004. Programa ***Cisco Networking Academy CCNA*** - 4 Módulos - Colegio Universitario Fermín toro. Barquisimeto - Edo. Lara Duración 1 año.

Año 2001. ***Ing. en Computación*** Egresado de la Universidad Fermín Toro. Barquisimeto - Edo. Lara. Duración 5 años

Cursos Realizados

- Año 2008 - ***Lenguaje de alto nivel- Java*** - 20 Horas dictado por UCLA – Barquisimeto Edo Lara.
- Año 2008 – ***Modelado de Software (UML, Metodologías de desarrollo, RUP, XP, Etc) –*** 20 Horas dictado por UCLA - Barquisimeto Edo Lara.
- Año 2007 - ***Administración de Bases de Datos Sql Server 2000*** - 32 Horas dictado por Infoserv c.a.
- Año 2006 - ***Curso Actualización de la Herramienta Case Inteligente Genexus 9.0*** - 24 Horas dictado por Maia Shuster Instructor Genexus Senior – Artech Consultores Montevideo-Uruguay instalaciones de Intercable Corporación Telemic.
- Año 2004 - ***Curso Oracle 9i*** - Administración y Nuevas Características, 24 Horas Gedica - Training Consulting en las instalaciones de Intercable Corporación Telemic.
- Año 2004 - Curso ***Java Development bajo Oracle 8.0***, 24 horas Gedica – Training - Consulting en las instalaciones de InterCable Corporación Telemic.
- Año 2004 - Curso de la herramienta ***Genexus 8.0.***
- Año 2003 - ***Administración de Microsoft Sql Server 2000***, 24 horas dictado por Global Knowledge.
- Año 2003 - ***English as Second Language***, Nivel I y II, en la Universidad Simón Bolívar(Asesores Integrales) – Núcleo Barquisimeto – Edo. Lara.
- Año 2002 - Diseño de Reportes Utilizando ***Oracle Report 6i***, 24 horas - Gedica Training Consulting.

- Año 2001 - ***English Conversation I*** - Westminster, Duración 120 Horas
- Año 2001 - ***Técnicas Avanzadas de Personalización en CRM – Pivotal - eRelationship*** – Duración 16 Horas - Grupo Lanka

Experiencia Laboral

- Año 2008 - ***Ingeniero de Proyectos Senior*** perteneciente a la Gerencia de proyectos de Software en la Vice-Presidencia de Sistemas, a Nivel Nacional de ***Inter*** Corporación Telemic. ***Trabajo Actual***. (Desde 16/05/2001).

- Año 2001-***Ingeniero de Proyectos de Software*** perteneciente a la Gerencia de proyectos de Software en la Vice-Presidencia de Sistemas, a Nivel Nacional de ***InterCable*** Corporación Telemic. ***Trabajo Actual***. (Desde 16/05/2001).
  Trabajo desempeñado: Diseño, Desarrollo de Software, Implementación, Mantenimiento y soporte para el Call Center de Intercable a nivel Nacional, específicamente módulo ***CRM*** (Atención al cliente telefónica - Televentas - Telecobranzas - HelpDesk - Alarmas - Mail Center - Integración con Central Telefónica –***CTI***) con la herramienta ***CRM Pivotal*** y ***SqlServer.***
  Migración del módulo ***CRM InBound – OutBound - CTI*** a la herramienta Case ***ERP Genexus*** generado en Java bajo Oracle, Módulo de Ventas de Calle, Servicios de Reconexión, Contratos, Facturas, atención al cliente, Llamadas de Bienvenida, etc.

- Año 2000-***Pasante*** en el Departamento de Sistemas en Licorerías Unidas S.A - Seagram). Duración: 4 meses.
  Trabajo desempeñado: Desarrollo de una Aplicación de Software para el departamento de costos de la empresa utilizando la herramienta Visual Basic, Access y Cristal Report .Básicamente la aplicación utilizaba comandos FTP para conectarse con el servidor principal de le empresa (AS400) y ejecutar una serie de programas en este, para luego realizar una serie de cálculos mensuales de producción de la empresa y emitir una serie de reportes usados en el departamento de Costos de la empresa.