

LA INGENIERÍA DE REQUISITOS EN LOS MÉTODOS DE DESARROLLOS
AGILES

SERGIO ANDRÉS MOSQUERA MORENO

CORPORACIÓN UIVERSITARIA ADVENTISTA



FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
MEDELLÍN
2015

LA INGENIERÍA DE REQUISITOS EN LOS MÉTODOS DE DESARROLLOS
AGILES

SERGIO ANDRÉS MOSQUERA MORENO

CORPORACIÓN UNIVERSITARIA ADVENTISTA



Trabajo Para Optar Al Título De Ingeniero De Sistemas

Asesor

Walter Hugo Arboleda Mazo

PROGRAMA DE INGENIERÍA DE SISTEMAS
MEDELLÍN
2015



CORPORACIÓN UNIVERSITARIA ADVENTISTA

FACULTAD DE INGENIERÍA

CENTRO DE INVESTIGACIONES

NOTA DE ACEPTACIÓN

Los suscritos miembros de la comisión Asesora del Proyecto de Grado: **"La Ingeniería de Requisitos en los Métodos de Desarrollo Agiles"**, elaborado por el estudiante SERGIO ANDRÉS MOSQUERA MORENO, del programa de Ingeniería de Sistemas, nos permitimos conceptuar que éste cumple con los criterios teóricos y metodológicos exigidos por la Facultad de Ingeniería y por lo tanto se declara como:

Aprobado

Medellín, Junio 03 de 2015

Mg (c). Walter Arboleda
Presidente

Mg. Jhon Niño Manrique
Secretario

Mg (c). Walter Arboleda
Vocal

Sergio Andrés Mosquera Moreno
Estudiante

Personería Jurídica según Resolución del Ministerio de Educación No. 8529 del 6 de junio de 1983 / NIT 860.403.751-3

Cra. 84 No. 33AA-1 PBX. 250 83 28 Fax. 250 79 48 Medellín <http://www.unac.edu.co>

DEDICATORIA

En esta etapa que es crucial en mi vida como la es el estudio he pasado por momentos difíciles y otros de victorias por la gracia del señor que siempre está allí para enseñarme lecciones de vida.

Por eso esta dedicación es primeramente para Dios quien me ha permitido llegar hasta aquí a pesar de las adversidades y por las bendiciones que ha derramado sobre mí para poder ir escalando cada vez más hacia el éxito.

A mis padres que me han apoyado en cada etapa de este proceso de mi vida.

AGRADECIMIENTOS

A Dios por guiar e iluminar mi camino y por la fortaleza que me ha dado para seguir adelante.

A mis padres por su amor por su comprensión y por ese apoyo incondicional que me brindan.

A mi asesor, Walter Hugo Arboleda Mazo por su orientación y su aporte en mi formación.

A la Corporación Universitaria Adventista por abrirme sus puertas y por el compromiso que ha tenido con mi crecimiento académico.

Contenido

LA INGENIERÍA DE REQUISITOS EN LOS MÉTODOS DE DESARROLLOS AGILES ..	viii
RESUMEN	viii
Capítulo 1	1
1. Introducción	1
1.1 Justificación.....	2
1.2.1 Objetivo General	3
1.2.1 Objetivos específicos	3
1.3 Metodología de trabajo	3
1.4 Investigación descriptiva.....	3
1.5 Diseño Bibliográfico.....	3
1.6 Población y muestra	4
1.7 Diseño metodológico.	4
CAPITULO 2.....	5
2. La Ingeniería de requisitos	5
2.1 Fases en la Ingeniería de Requisitos.....	6
2.1.1 Elicitación de requisitos ¿Cómo se trabaja?.....	7
2.1.1.1 Técnicas de elicitación de Requisitos	8
2.2 Especificación de requisitos.....	11
2.2.1 Dificultad para definir los requerimientos.....	11
2.2.2 Clasificación de requerimientos de sistemas de software.....	12
2.3 Verificación y validación de requisitos	13
2.4 Gestión de requisitos	15
2.4.1 Gestión de cambio.....	16
CAPITULO 3.....	18
3. La ingeniería de requisitos en los métodos ágiles	18
3.1 Acercamiento a las metodologías Ágiles	18
3.1.1 Programación Extrema.....	20
3.1.1.1 Fundamentos de Extreme Programming	21
3.1.1.2 Modelo de requerimientos y Extreme Programming.....	22
3.1.1.2.1 Historias de usuario.....	22
3.1.1.2.2 Task Card (Tareas De Ingeniería).	23
3.1.1.2.3 Tarjetas CRC (Clase - Responsabilidad – Colaborador).....	24

3.1.2	Scrum	25
3.1.2.1	Modelo de requerimientos y Scrum	26
3.1.2.2	Product Backlog.....	26
3.1.2.3	Sprint Backlog.....	27
3.2	Beneficios de Scrum según la industria	27
3.2.1	Ceiba Software House SAS.....	27
CAPITULO 4.....		29
4.	Resultados	29
4.1	Metodologías utilizadas para el desarrollo Ágil de software	29
4.2	Técnicas utilizadas para la Elicitacion de requisitos	29
4.3	Obstáculos para la Adopción Ágil.....	30
4.4	Beneficios que se obtendrían al aplicar de forma adecuada metodologías ágiles en proyectos de desarrollo.....	31
4.5	Técnicas usadas para la priorización de Historias de usuario.	32
4.6	Empresas encuestadas dedicadas al desarrollo de software para el sector salud en Medellin.....	36
4.7	Encuesta ingeniería de requisitos empresas dedicadas al desarrollo de software en el sector de la salud en Medellin.	37
4.8	Conclusiones.....	41
4.9	Trabajos futuros.....	42
4.10	Referencias Bibliográficas.....	43

INDICE DE ILUSTRACIONES

<i>Ilustración 1</i> Proceso de Ingeniería de requisitos Fuente: [10].....	6
<i>Ilustración 2</i> Importancia de una buena gestión de requisitos Fuente [4]	7
<i>Ilustración 3</i> Proceso de ingeniería de requerimientos Fuente [15]	8
<i>Ilustración 4</i> Definiendo los requisitos[24]	12
<i>Ilustración 5</i> Tipos de requerimientos no funcionales[20].....	13
<i>Ilustración 6</i> Gestión de Seguridad funcional [28]	14
<i>Ilustración 7</i> Proceso de gestión de cambios[29]	16
<i>Ilustración 8</i> Requerimientos generales. Fuente [5]	19
<i>Ilustración 9</i> Tarjeta de CRC [36].....	24
<i>Ilustración 10</i> Esquema General Scrum [46]	25
<i>Ilustración 11</i> Product Backlog Ítems[47].....	26
<i>Ilustración 12</i> Sprint Backlog[48]	27
<i>Ilustración 13</i> Metodologías ágiles Fuente: Propia Autoría	29
<i>Ilustración 14</i> Técnicas para la Elicitación de Requisitos Fuente: Propia Autoría.....	30
<i>Ilustración 15</i> Obstáculos para la adopción Ágil Fuente: Propia Autoría	31
<i>Ilustración 16</i> Beneficios aplicación de metodologías Ágiles Fuente: Propia Autoría.....	32
<i>Ilustración 17</i> Técnicas de Priorización Fuente: Propia Autoría	32
<i>Ilustración 18</i> Información proceso de elicitation Requisitos Fuente: Propia Autoría.....	33
<i>Ilustración 19</i> Preparación de sesiones de Elicitación Fuente: Propia Autoría	33
<i>Ilustración 20</i> Identificación y revisión de los objetivos del sistema Fuente: Propia Autoría.....	34
<i>Ilustración 21</i> Requisitos de Información presentadas por el usuario Fuente: Propia Autoría.....	34
<i>Ilustración 22</i> Revisión requisitos funcionales por el usuario Fuente: Propia Autoría.....	35
<i>Ilustración 23</i> Revisión requisitos no funcionales Fuente: Propia Autoría.....	35
<i>Ilustración 24</i> Priorización de los objetivos y requisitos Fuente: Propia Autoría	35
<i>Ilustración 25</i> Realización de historias de Usuarios y Product Backlog Fuente: Propia Autoría.....	36
<i>Ilustración 26</i> Encuesta Empresas Pt 1 Fuente: Propia Autoría.....	37
<i>Ilustración 27</i> Encuesta empresas Pt2 Fuente: Propia Autoría	38
<i>Ilustración 28</i> Encuesta empresas Pt3 Fuente: Propia Autoría	39
<i>Ilustración 29</i> Encuesta Empresas Pt4 Fuente: Propia Autoría.....	40

INDICE DE TABLAS

<i>Tabla 1 Elicitacion de requisitos funcionales Fuente: Propia Autoría</i>	12
<i>Tabla 2 Principios del agilísimo [15]</i>	20
<i>Tabla 3 Historias de Usuario Fuente [35]</i>	22
<i>Tabla 4 Caso de prueba aceptación[35]</i>	23
<i>Tabla 5 Tarea de Ingeniería [35]</i>	23
<i>Tabla 6 Empresas Encuestadas Fuente: Propia Autoría</i>	36

LA INGENIERÍA DE REQUISITOS EN LOS MÉTODOS DE DESARROLLOS AGILES

RESUMEN

Este proyecto de grado investiga acerca de la ingeniería de requisitos y los métodos de desarrollo ágil con el fin de realizar un estado del arte donde se pueda describir la ingeniería de requerimiento, como ese proceso de descubrimiento, refinamiento y, análisis de lo que se desea construir. En este proceso juega un papel muy importante el cliente al igual que el analista. Cuando se realiza un proyecto de desarrollo, es de suma importancia dejar claramente definidos los requerimientos del software de forma consistente y compacta. Esta tarea es difícil, principalmente por que consiste en la traducción de unas ideas vagas y convertirlas en necesidades de software y en un conjunto concreto de funciones y restricciones. Además el analista debe extraer información dialogando con muchas personas y cada una de esas personas se expresan de una forma muy diferente. Las metodologías de desarrollo ágil permiten establecer cómo se va a desarrollar el sistema desde un principio. Permite un trabajo activo para eliminar la complejidad del sistema.

Palabras Clave: Ingeniería, calidad, estado del arte, metodologías, desarrollo, ágil, ingeniería de requisitos.

Abstract

This degree project investigates about requirements engineering and agile development methods in order to make a State of the art where you can describe engineering requirement, as that process of discovery, refinement, and analysis of what you want to build. In this process plays a very important role the customer as well as the analyst. While a development project, is of utmost importance to leave clearly defined software requirements consistently and compact. This task is difficult, mainly because it consists in the translation of vague ideas and turn them into software needs and a specific set of features and restrictions. In addition the analyst must extract information dialogue with many people and every one of these people express themselves in a very different way. Agile development methodologies allow to establish how is going to develop the system from the very beginning. It allows a work...

Keywords: Engineering, quality, State of the art methodologies, development, agile, requirements engineering.

Capítulo 1

1. Introducción

A través de los años se ha podido constatar que los requisitos son la pieza fundamental en un proyecto de desarrollo de software, ya que marca el punto de partida para actividades como la planeación, estimación de tiempo y costos. La siguiente investigación se refiere a la Ingeniería de Requisitos esta trata de los principios, métodos, técnicas y artefactos que permiten descubrir, documentar y mantener los requisitos para sistemas basados en computadora. Para poder desarrollar un software de calidad que tenga un rendimiento eficiente, que sea fácil de manejar y que ahorre recursos es necesario tener presente la ingeniería de requisitos ya que este brinda calidad y mejoramiento en los procesos dentro del desarrollo. Muchos han dejado de lado la documentación de sus proyectos (Ingeniería de requerimientos) esto se debe a que han interpretado mal el concepto de metodologías ágiles lo cual hace que solo se concentren en desarrollar sin trabajar de la mano de esta disciplina y como resultado de esto muchos proyectos con el pasar del tiempo quedan obsoletos ya que no se pueden modificar, ni agregar nuevos módulos. La investigación de esta problemática se realizó por el interés de establecer los factores que generan dificultad a la hora de aplicar la ingeniería de requisitos en los métodos de desarrollo ágil mediante un enfoque cualitativo. Para la realización de este proyecto se tomara como instrumento la encuesta, también está apoyada en la observación directa y el estado del arte de la ingeniería de requisitos, de esta forma se recopiló la información necesaria.

1.1 Justificación

A la hora de construir un proyecto de software, la parte más difícil de establecer son los requisitos. Porque en ese momento se empieza a pensar cual es la mejor opción para obtener el aplicativo que se desea. Roger Pressman en [1] argumenta que con el pasar de los años ha crecido el campo de la tecnología y la comunicación, con este crecimiento ha aumentado sustancialmente el campo de desarrollo de software para suplir todas las necesidades de las maquinas existentes en el mercado. Gracias a este crecimiento nació una disciplina llamada ingeniería de requisitos, la cual se enfocó en estándares que conllevan a tener productos de calidad. En la actualidad muchos proyectos de desarrollo de software fracasan o quedan obsoletos con el pasar del tiempo, porque no hacen un análisis correcto y no definen en los requisitos lo que realmente pide el usuario, y esto hace que no se le dé una solución al problema propuesto. Alonso Toro & Lorena Cardona En [2] argumentan que la ingeniería de requisitos se ha convertido en una base fundamental para garantizar que la construcción y el mantenimiento de productos de software sean productos eficientes y de calidad, por esta razón es importante que el desarrollador tenga conocimientos acerca de la ingeniería de requisitos y que también comprenda cómo aplicar esta disciplina en las metodologías ágiles. Con esta investigación y por medio de la realización de una encuesta se busca saber cuál es la metodología más utilizada en las empresas que se dedican a desarrollar software para el sector de la salud en Medellin y cuáles son los beneficios que se obtienen al trabajar con dichas metodologías. Además se sabrá cuáles son los factores que impiden utilizarlas y la relación que tiene la ingeniería de requisitos y los métodos de desarrollo ágil para la industria del software de la salud en Medellin. De igual forma se desea mostrar la importancia de la ingeniería de requisitos ya que esta es la base para que los desarrolladores de software logren que sus proyectos sean eficientes y eficaces, también se hará un estado del arte del mismo para saber el avance que se ha tenido en los últimos años y aproximarnos a los factores que hacen que los desarrolladores no apliquen buenas practicas teniendo en cuenta que con la ingeniería de requisitos se busca calidad.

1.2 Objetivos

A continuación se presentaran los objetivos planteados para el desarrollo y focalización del trabajo de investigación.

1.2.1 Objetivo General

Establecer factores que generan dificultad en el uso de la ingeniería de requisitos en los métodos de desarrollo ágil en empresas dedicadas al desarrollo de software para el sector salud en Medellín mediante un enfoque mixto para canalizarlos de tal forma que se conviertan en una fortaleza para la comunidad.

1.2.1 Objetivos específicos

- Saber cuáles son la metodología más utilizada en las empresas dedicadas al desarrollo de software para el sector salud en Medellín y cuáles son los beneficios que se obtienen al trabajar con dichas metodologías.
- Presentar cuales han sido los obstáculos en las empresas para adoptar metodologías ágiles de trabajo.

1.3 Metodología de trabajo

1.4 Investigación descriptiva

Esta investigación describirá los problemas que han tenido las empresas dedicadas al desarrollo de software para el sector salud en Medellín buscando adoptar metodologías ágiles e ingeniería de requisitos. Para poder dar esta descripción se les hará una encuesta a dichas empresas.

Diseño de la investigación

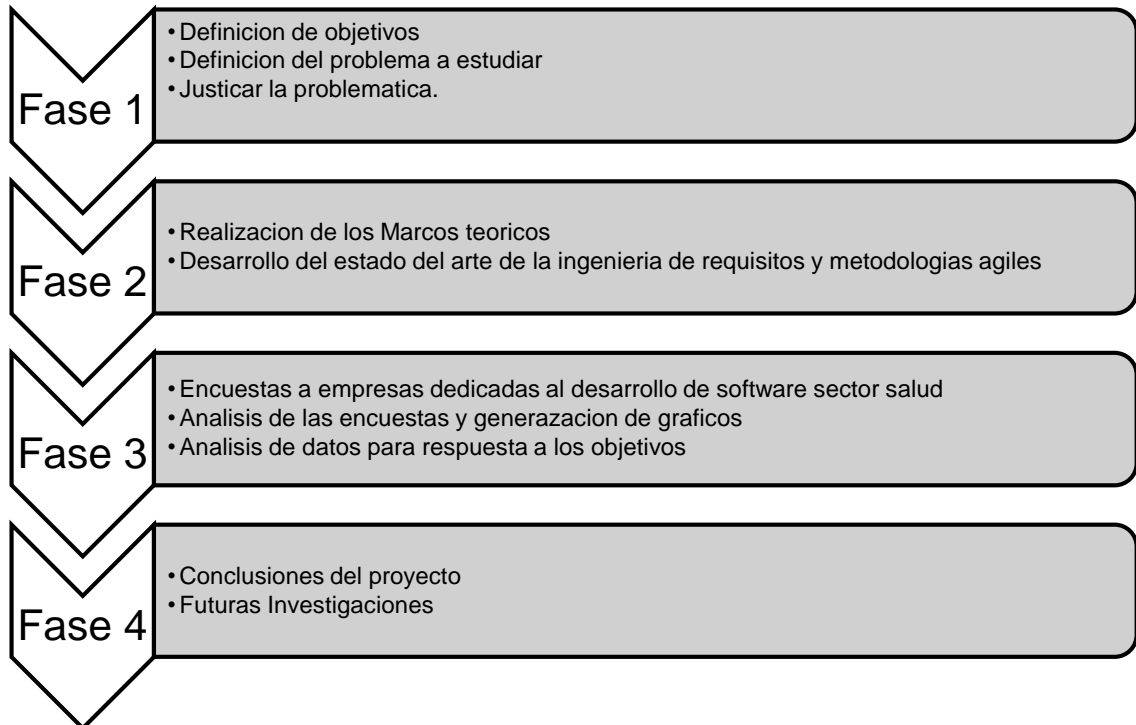
1.5 Diseño Bibliográfico

En esta tesis se hará un estado del arte acerca de la ingeniería de requisitos y metodologías ágiles por lo cual es indispensable tener referencias de datos secundarios para saber que se ha dicho y como la opinión de otros autores enriquecen nuestro proyecto. Estas bibliografías serán expuestas con el formato IEEE y generadas a través de Mendeley.

1.6 Población y muestra

En el desarrollo de este proyecto la población serán las empresas dedicadas al desarrollo de software para el sector salud en Medellín y la muestra será todas las empresas que se logre encuestar.

1.7 Diseño metodológico.



CAPITULO 2

Introducción

En el presente capítulo se expone la ingeniería de requisitos. En este se presentan las fases que componen el proceso para hacer una correcta obtención de los requerimientos. Este proceso está compuesto por varias actividades como son la Elicitación, especificación, validación y verificación de requisitos.

2. La Ingeniería de requisitos.

La ingeniería de requisitos, es un conjunto de procesos, tareas y técnicas que permiten la definición y gestión de los requisitos de un producto de un modo sistemático. María Torres en [3] afirma que estos facilitan los mecanismos adecuados para comprender las necesidades del cliente, analizándolas, confirmando la viabilidad, negociando una solución razonable, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. Paul Dunne en [4] define que estos requisitos tienen una descripción diferente dependiendo de quién esté hablando en ese momento; para un desarrollador es una definición matemática formal y detallada de una función del sistema mientras que para un gerente es una descripción abstracta de los servicios o limitaciones para el sistema. Roger Pressman en [5] afirma que la ingeniería de requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajaran. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, que es lo que el cliente quiere y como interactuaran los usuarios finales con el software.

La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administra los requisitos conforme estos se transforman en un proceso operacional. El proceso de la ingeniería de requisitos se lleva a cabo de siete funciones: Inicio, obtención, elaboración, negociación, especificación, validación y función. Algunas de estas funciones ocurren en paralelo y todas deben adaptarse a las necesidades del proyecto, todas están sujetas a lo que el cliente quiera y todas sirven para establecer una base sólida respecto al diseño y la construcción de lo que obtendrá el cliente. Sai Ganesh en [6] argumenta que los requisitos son declaraciones de lo que el sistema debe hacer, cómo debe comportarse, las propiedades que debe exhibir, las cualidades que debe poseer, y las limitaciones que el sistema y su desarrollo deben satisfacer. Anthony Hall & Pressman en [7],[1] definen los requisitos como una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo, condición o capacidad que se debe cumplir o lo que debe hacer el sistema para satisfacer un

contrato, normal, especificación u otro documento formalmente impuesto, una representación documentada de una condición o capacidad.

2.1 Fases en la Ingeniería de Requisitos

Natalia Valeria A. en [8] afirma que la ingeniería de requisitos ayuda a descubrir cuáles son las metas, necesidades y expectativas que se tienen para iniciar un proyecto de desarrollo de software. La ingeniería de requerimientos está formada por una serie de procesos, Adelaida Ramírez en [9],[10] expone que en estos procesos no existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad de los resultados. La definición de ingeniería de requisitos no es del todo clara ya que muchos autores generan clases que son entendidas por otros autores como fases posteriores dentro de las posibles etapas de la ingeniería de requisitos. la propuesta que ha tenido mayor acogida es la establecida por Lowe & hall [11],[12] en ella el proceso de trabajos con los requisitos está compuesta por las siguientes actividades:

- Elicitación de requisitos.
- Especificación de requisitos.
- Verificación y validación de requisitos

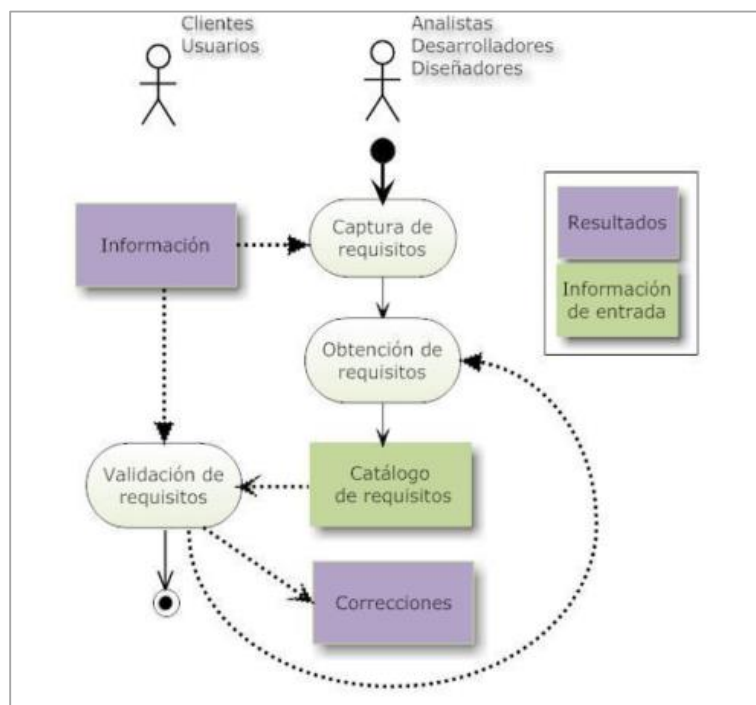


Ilustración 1 Proceso de Ingeniería de requisitos Fuente: [10]

2.1.1 Elicitación de requisitos ¿Cómo se trabaja?

Carnegie University [13] define la elicitation de requisitos como el proceso de descubrir, revisar, documentar, comprender las necesidades y limitaciones del usuario para el sistema, es una ciencia y disciplina que resuelve un problema en el mundo real, este se ocupa de establecer y documentar los requisitos del software. Markus Namies en [14] afirma que la elicitation de requisitos es la llave que posibilita el desarrollo de proyectos de software con calidad; ya que este tiene un alto impacto en el diseño y en las fases que componen el ciclo de vida del producto, Implementarla de una forma adecuada hace que se reduzcan las correcciones en los requisitos teniendo en cuenta que la ER¹ es el inicio de un proyecto de desarrollo de software por esta razón se convierte en la etapa más crítica y de alto impacto dentro del mismo. Los errores y defectos del software son el resultado de una carente gestión y definición de requisitos, se podría decir que la gran mayoría de los problemas encontrados dentro de un proyecto de software se pudieron haber evitados si se fuera hecho una buena abstracción de lo que realmente el cliente esperaba del proyecto.

Ceiba Software House SAS al momento de iniciar un proyecto de software usa las técnicas de Entrevista, Lluvia de ideas, Desarrollo de Prototipos (Demos) para poder tener un panorama más claro de lo que el cliente desea obtener. Muchos dicen "considerar"² la ingeniería de requisitos como la primera etapa, yo la definiría como el Alfa, Esta es indispensable e irremplazable para desarrollar un proyecto de software de calidad y con la garantía de que este cumpla las necesidades del usuario final. Ya que si se hace una buena gestión de los requisitos se ahorrara dinero ya que en el futuro no se tendrá que hacer modificaciones costosas.

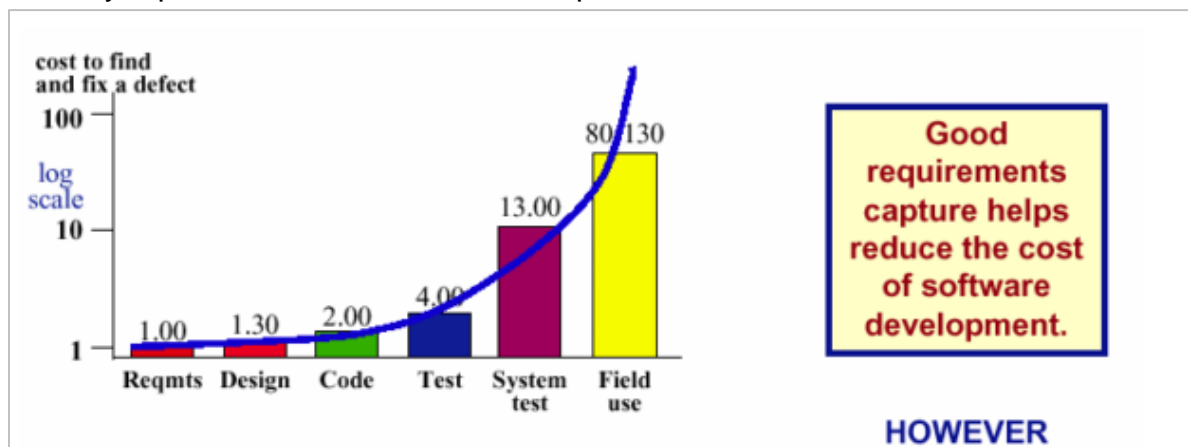


Ilustración 2 Importancia de una buena gestión de requisitos Fuente [4]

¹ Elicitación de requisitos

² Cuando hablamos de considerar se apunta a reflexionar para poder escoger la opción correcta

Natalia Adriano en [8] argumenta que durante la elicitación de requerimientos es necesario identificar todas y cada una de las personas afectadas por este proceso. Esta primera actividad es importante ya que es necesaria la elicitación desde múltiples puntos de vista porque una sola persona no sabe todo. Durante este proceso de identificación temprana de stakeholders³ es necesario tener en cuenta al menos a los líderes de proyectos, a los ingenieros de requerimientos de software, especialistas en documentación y a los potenciales usuarios. A continuación se mostrara una ilustración donde se ve reflejada el proceso de la ingeniería de requisitos.

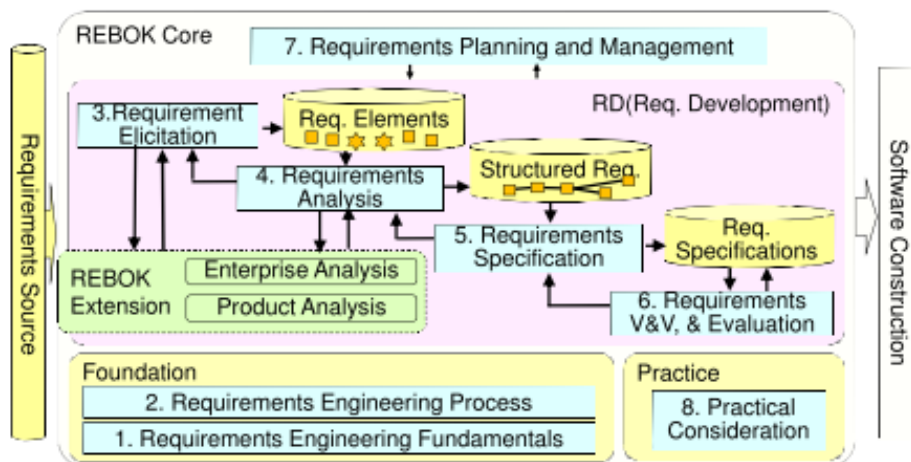


Ilustración 3 Proceso de ingeniería de requerimientos Fuente [15]

2.1.1.1 Técnicas de elicitación de Requisitos

Linda Westfall en [16] afirma que muchas técnicas se pueden usar para obtener los requisitos incluyendo entrevistas con los interesados, grupos focales, observaciones de los procesos de trabajo actuales, cuestionarios y encuestas, análisis de productos de la competencia, y la evaluación comparativa de las prácticas de la industria. Para reducir las dificultades, se presentara una breve descripción de alguna de las técnicas más conocidas para la elicitación de requisitos.

³ Este término se utiliza para referirse a cualquier persona que tiene influencia directa o indirecta sobre los requisitos del sistema. Entre los stakeholders se encuentran los usuarios finales que interactúan con el sistema y todos aquellos en la organización que se verán afectados por dicho sistema.

2.1.1.2 Entrevista

En el libro Ingeniería de Software de R. Pressman [5] afirma que las entrevistas dan información específica de un individuo, para la correcta aplicación se requiere habilidades de comunicación social, conocimiento de tácticas de entrevista y buena capacidad para escuchar.

Fases para la realización de la entrevista

Identificación de personal

Roger Pressman [5],[17] manifiesta que se debe escoger con minuciosidad a las personas que se les realizara la entrevista, ya que si esta no tiene el conocimiento adecuado de lo que se realizara, la elicitación generara datos incorrectos.

La primera persona a la que se debe realizarle la entrevista es a quien patrocina la realización del software, luego se debe revisar el organigrama de la empresa que usara el software para así seleccionar a alguien que tenga conocimiento de los roles que cumple cada empleado.

Preparación

Roger Pressman [18] afirma que el entrevistador debe hacer arreglos con respecto al cronograma de la entrevista, las entrevistas deben ser planeadas con respecto al tiempo que se llevara a cabo y con un tiempo adicional para destinarle al entrevistado si quiere dar comentarios extras y para la realización de un resumen de dicha entrevista. El entrevistador debe fijar con qué objetivo es realizada la entrevista además debe enviar cualquier información relevante con tiempo suficiente de antelación. Este debe preparar sus preguntas de tipo (abierta y/o cerrada) y debe hacerlas de acuerdo al rol que cumpla la persona entrevistada.

Desarrollo de la entrevista

En el instante que comience la entrevista el entrevistador debe explicar el objetivo y las metas de la entrevista, las preguntas se realizaran teniendo en cuenta el orden ya acordado y tratando que la conversación no se salga de contexto. El entrevistador debe escuchar con detenimiento lo que el entrevistado diga y no debe adelantarse a la respuesta, dejar que el entrevistado termina la última frase antes de seguir con la próxima pregunta. Se debe permitir que el entrevistado haga preguntas pero que estas no se salgan del tema y es de suma importancia no olvidar el tiempo de duración de la entrevista. N. Valeria en [8] Afirma que antes de comenzar con una nueva entrevista se deberá realizar un resumen de la entrevista

realizada, enviar una copia al entrevistado y pedir confirmación del resumen realizado.

2.1.1.3 Encuesta

La encuesta es una búsqueda sistemática de información en la que el investigador pregunta a los investigados sobre los datos que desea obtener, y posteriormente reúne estos datos individuales para obtener durante la evaluación datos agregados. Con la encuesta se trata de "obtener, de manera sistemática y ordenada, información sobre las variables que intervienen en una investigación, sobre una población o muestra determinada. Esta información hace referencia a lo que las personas son, hacen, piensan, opinan, sienten, esperan, desean, quieren o desaprueban. Amador Duran en [19] expresa que las encuestas deben elicitarse el punto de vista del cliente, sus metas, procesos, problemas, mejoras sugeridas y estructura lógica del producto, entre otros. Una de las ventajas que provee esta técnica es que las encuestas pueden ser reutilizadas en nuevos proyectos. Natalia Adriano en [8] afirma que una encuesta podría ser tabulada completamente y ser re-expresado en una serie de diferentes diagramas. Las tablas y los diagramas deben ser validados por el cliente. Como resultado se obtiene una manera rápida y efectiva de obtener una caracterización detallada de la perspectiva del cliente.

2.1.1.4 Lluvia de ideas

Roger Pressman en [5] dice que esta es una de las técnicas más simples para la generación de ideas, consiste en la reunión de un grupo de personas asociadas con el producto a desarrollar y estas expresan sus ideas con total libertad. Generalmente estas reuniones son realizadas con un número de 4 a 10 personas, una de ellas será escogida como lidere para comenzar o dar la palabra pero no para restringir la opinión de nadie. Esta tiene una serie de reglas que se mostraran a continuación.

- No está permitida la crítica y el debate
- Generar la mayor cantidad de ideas posibles
- Cambiar y combinar las ideas
- Reducción de ideas
- Eliminar ideas que no estén dentro del contexto
- Agrupar ideas similares
- Priorizar las ideas restantes

2.2 Especificación de requisitos

La especificación de requisitos consiste en determinar y fijar las necesidades precisas de los usuarios sobre la parte del sistema de información a desarrollar. Alguno de los problemas que surgen durante el proceso de ingeniería de requisitos es el resultado de no hacer una clara separación entre los niveles de descripción como lo son los requerimientos de usuario que son utilizados para designar los requerimientos abstractos de alto nivel y los requerimientos de sistemas para estructurar detalladamente lo que el sistema deben hacer.

Requerimientos de usuario.

Somerville en [22] menciona que los requerimientos de usuario son declaraciones en lenguaje natural, Estos conforman un diagramas de los servicios que se espera que el sistema proporcione.

Requerimientos del sistema.

Somerville en [20] manifiesta que estos se establecen con una directa especificación de las funciones, servicios y restricciones operativas del sistema, este debe ser preciso, debe definir exactamente qué es lo que se va implementar.

2.2.1 Dificultad para definir los requerimientos

M. Sandoval & M. Adilia en [22] declaran que los requerimientos no son explícitos y que además estos provienen de muchas fuentes, alguno de ellos son difíciles de expresar en palabras y esto hace que se creen ambigüedades.

M. Gómez Fuentes en [23] afirma que para realizar un buen proyecto es esencial tener una especificación completa de los requerimientos, Independiente mente si se tiene un buen diseño o una buena codificación. Un sistema pobremente especificado decepcionara al usuario y hará fracasar el desarrollo, la forma de especificar tiene que ver con la calidad de la solución. Cuando se tiene una especificación incompleta las consecuencias se padecen en la calidad, oportunidad y completitud del software que se desea.



Ilustración 4 Definiendo los requisitos[24]

2.2.2 Clasificación de requerimientos de sistemas de software

Requisitos funcionales

Feng Chen en [25] expresa que la obtención de requisitos produce principalmente requisitos funcionales porque estos representan las funcionalidades que el sistema debe cubrir, mediante la descripción de casos de uso los actores utilizan los servicios proporcionados por el sistema. Cada requisito funcional llega a identificarse con el evento de activación, Pre y post condiciones, así como los pasos que componen el caso de uso junto con sus excepciones. V. Valdez en [26] menciona que los requerimientos de software son las necesidades de los Stakeholders que requiere que el Sistema deba de cumplir de manera Satisfactoria. Son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas

ID. Requisitos	Nombre del requisito	Descripción del requisito	Rol
RF-001	[XX]	[XX]	[XX]

Tabla 1 Elicitacion de requisitos funcionales Fuente: Propia Autoría

Requisitos no funcionales

Feng Chen en [25] afirma que la captura y el tratamiento de NFR siguen siendo un reto tanto para el mundo académico y los profesionales. La profesión software ha exagerado la funcionalidad en el costo de NFR. Los requerimientos no funcionales, son aquellos que no van dirigidos a las funciones específicas del sistema, si no a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De igual forma, definen las restricciones del sistema como capacidad de los dispositivos de entrada/salida y las representaciones de datos que utilizan en las interfaces del sistema. S. Gunda en [6] manifiesta que cualquier sistema que no ofrece un servicio fiable y medidas de seguridad contra las amenazas no se está considerando como un éxito. Estos requisitos forman la base de la calidad del sistema. Estos no solo se refieren al sistema software a desarrollar. Algunos de estos requerimientos pueden restringir el proceso que se debe utilizar para desarrollar el sistema, estos surgen de las necesidades del usuario, debido a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas software o hardware o a factores externos.

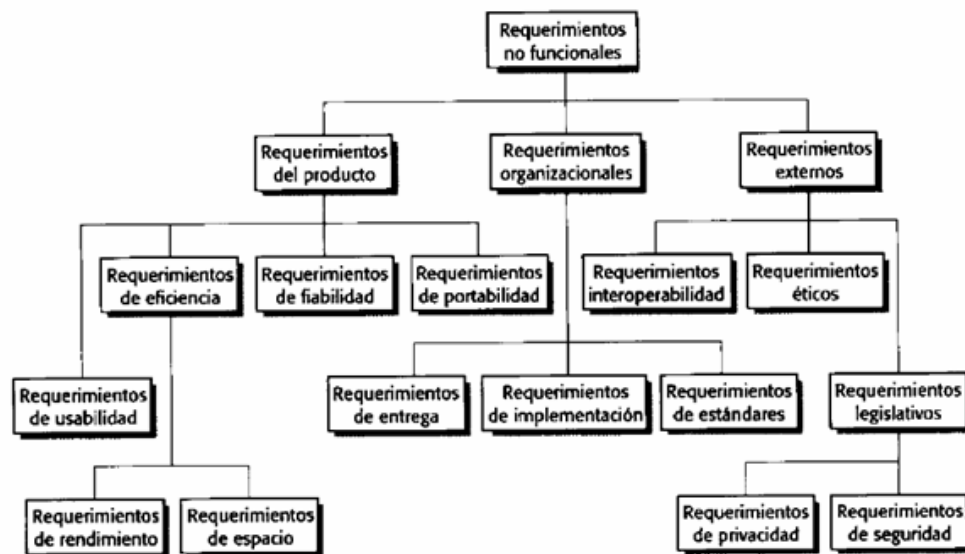


Ilustración 5 Tipos de requerimientos no funcionales[20]

2.3 Verificación y validación de requisitos

J. Drake en [27] declara la V&V como el proceso de análisis y comprobación que asegura que el software que se desarrolla cumple y está acorde con las necesidades de los clientes. Este es un proceso de ciclo de vida completo el cual inicia con las revisiones de los requerimientos y continúa con la revisión del diseño,

inspecciones del código hasta la prueba del producto. Existen actividades de verificación y validación en cada etapa del proceso de desarrollo del software. La V&V no son la misma cosa, aunque es muy fácil de confundirlas. La validación hace estudiar si en realidad se está construyendo el producto correcto en cambio que la verificación muestra si se está construyendo correctamente el producto deseado.

2.3.1 Verificación

Explicándolo de una forma más detallada la verificación implica comprobar que el software este de acuerdo con su especificación. Se debe comprobar que satisfaga los requerimientos funcionales y no funcionales y demuestra que el trabajo ha cumplido con los objetivos.

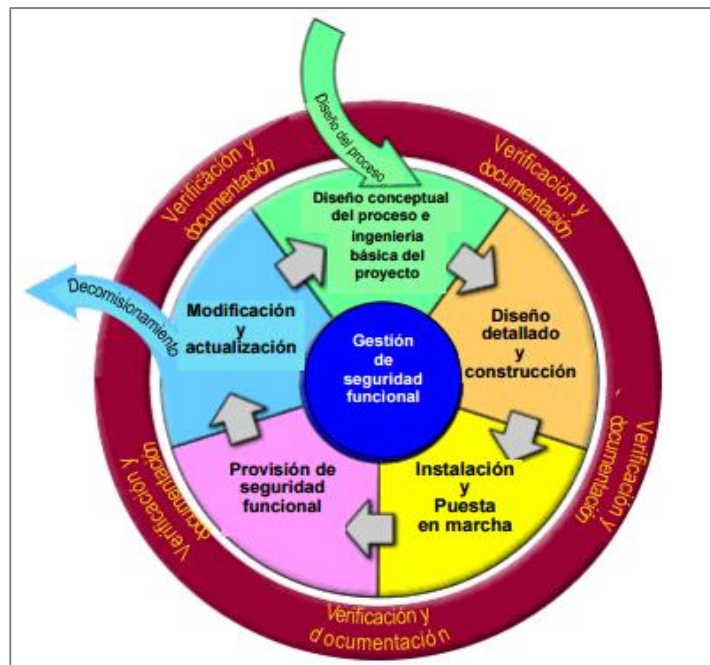


Ilustración 6 Gestión de Seguridad funcional [28]

La verificación se puede realizar mediante análisis, pruebas o una combinación de ambos. Las actividades podrían incluir:

- Revisión de documentos de todas las fases del ciclo de vida de seguridad para garantizar el cumplimiento con los objetivos y requisitos
- Revisión del diseño
- Pruebas de los productos diseñados para garantizar que funcionen de acuerdo con su especificación. Esto es especialmente valioso para

componentes modulares (tales como el código para un algoritmo de votante que se usará muchas veces).

- Pruebas de integración realizadas cuando se juntan diferentes partes del sistema.

INTECO⁴ en [29] menciona que en esta fase, el usuario final añade criterios de aceptación para cada requisito, además, apoya el hecho de que los requisitos han de ser correctos antes de que sean entregados a los diseñadores y desarrolladores. En este proceso la puerta de calidad es un punto por el que pasan cada uno de los requisitos antes de formar parte de la especificación. Una de las tareas de las puertas de calidad es asegurarse de que cada requisito cumple con el criterio que tiene asignado. Este criterio es una medida del requisito que le hace entendible y con capacidad para ser probado. Otra razón por la que el proyecto tiene puertas de calidad es para prevenir posibles fugas de requisitos. Los requisitos, algunas veces, aparecen en las especificaciones sin que nadie realmente sepa de donde vienen o qué valor añaden al producto. Asegurándose de que la única forma de que los requisitos entren a formar parte de las especificaciones sea a través de las puertas de calidad, el equipo del proyecto tiene un total control de los requisitos.

2.3.2 Validación

Somerville en [20] define la validación es un proceso general, su objetivo es asegurar que el sistema software satisfaga las expectativas del cliente. Va más allá de comprobar si el sistema está acorde con su especificación, para probar que el software hace lo que el usuario espera a diferencia de lo que se ha especificado. Es importante llevar a cabo la validación de los requerimientos del sistema de forma inicial. Es fácil cometer errores y omisiones durante la fase de análisis de requerimientos del sistema y, en tales casos, el software final no cumplirá las expectativas de los clientes. Sin embargo, en la realidad, la validación de los requerimientos no puede descubrir todos los problemas que presenta la aplicación. Algunos defectos en los requerimientos solo pueden descubrirse cuando la implementación del sistema es completa.

2.4 Gestión de requisitos

INTECO & J. Drake en [28],[27] nombran que es de suma importancia mostrar los requisitos para el sistema propuesto ya que es indispensable para que el equipo sepa las metas que deben lograr.

⁴ Instituto Nacional de Tecnologías de la Comunicación.

Sin los requisitos no se puede controlar la productividad, ni obtener datos que se adecuen a las pruebas, ni se puede predecir el tamaño y esfuerzo del proyecto es decir los requisitos escritos o previamente establecidos en definitiva son necesarios.

Por lo tanto, los objetivos principales del proceso de gestión de requisitos serán:

- Gestionar la recogida de requisitos
- Obtener la aprobación de los participantes del proyecto
- Gestionar los cambios (trazabilidad)

La gestión de requisitos es un proceso que se desarrolla a lo largo de toda la vida del producto.

2.4.1 Gestión de cambio

J.M Drake en [27] afirma que los requisitos cambian durante todo el ciclo de vida de desarrollo de un producto. Estos cambios deben controlarse y documentarse es decir hay que convivir con estos cambios y por lo tanto la gestión de cambio es la utilizada para tratar los mismos. El siguiente gráfico muestra el proceso de gestión de cambios con las actividades a llevar a cabo durante su desarrollo.

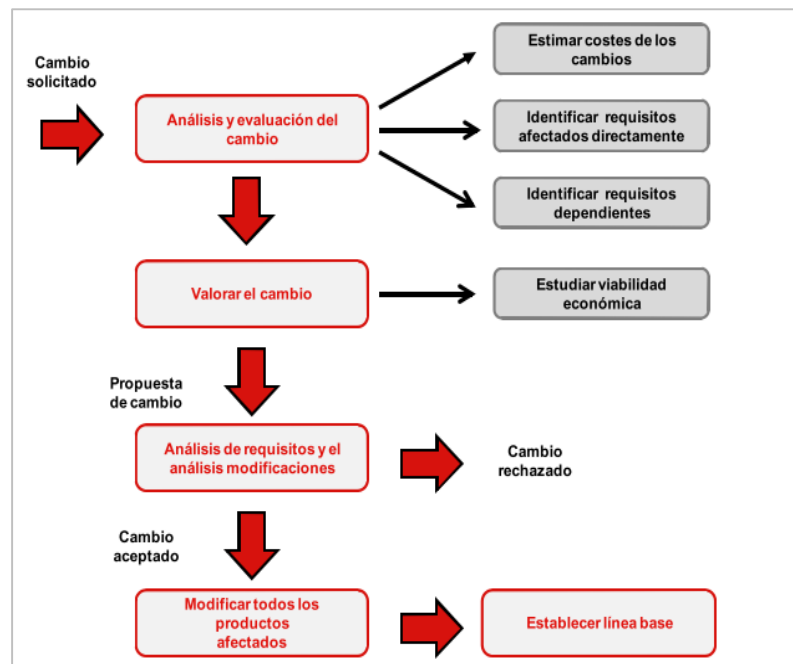


Ilustración 7 Proceso de gestión de cambios[29]

A continuación se descartaran tres importantes actividades dentro del proceso de gestión de cambios:

- **Evaluar el impacto**
La primera tarea a realizar tras recibir una petición de cambio es valorar el impacto del mismo. Para ello se deberá ir recorriendo todo el árbol de requisitos viendo como les afecta el cambio, y aquí es donde entra la trazabilidad de los requisitos.
- **Aceptación del cambio**
Una vez analizado el impacto del cambio, se debe tomar una decisión. Si se acepta el cambio, tras negociarlo con el cliente, se continuará con la actividad de implementar el cambio. En caso contrario, se deberá negociar con el cliente el siguiente paso a realizar.
- **Implementación del cambio**
Si se ha aceptado el cambio, hay que reflejar ese cambio en todos los productos que resulten afectados por dicho cambio.

CAPITULO 3

Introducción

En esta sección se muestra cómo las metodologías ágiles trabajan por medio de los artefactos para hacer la obtención de los requisitos dentro de proyectos de desarrollo de software; las metodologías ágiles a trabajar dentro de esta sesión serán Extreme Programming y Scrum.

3. La ingeniería de requisitos en los métodos ágiles

Mikio Aoyama & Takako Nakatami en [18] afirman que las tareas de elicitation, documentación y verificación de requisitos hacen parte de la ingeniería de requerimientos y permiten conocer la realidad de un proyecto de desarrollo, Pero son los cambios constantes que hacen que un proyecto sea más complejo a la hora de desarrollarlo. Edgar Serna en [9] argumenta que las metodologías ágiles ayudan a mejorar la atención a las modificaciones que los usuarios requieren para el sistema propuesto. Por esto se investiga sobre los métodos, técnicas y estrategias que permitan alcanzar resultados positivos en el desarrollo de software. En este proyecto de investigación se analizan las metodologías más utilizadas con una mirada en las etapas de gestión de requisitos dentro del ciclo de vida del software, donde se explican las prácticas, técnicas y artefactos que utilizan en las mencionadas etapas. Para ello se indago en documentos referente a la ingeniería de requisitos y metodologías ágiles para conocer estos procesos y comprender cuál es su esquema de trabajo.

3.1 Acercamiento a las metodologías Ágiles

Las metodologías ágiles es un tema naciente en la ingeniería del software y que ha venido tomando mucha fuerza. En los años 80 y principio 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, una garantía de calidad formalizada, la utilización de métodos de análisis y diseño soportados por herramientas CASE⁵, y procesos de software controlados y rigurosos.

⁵ son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

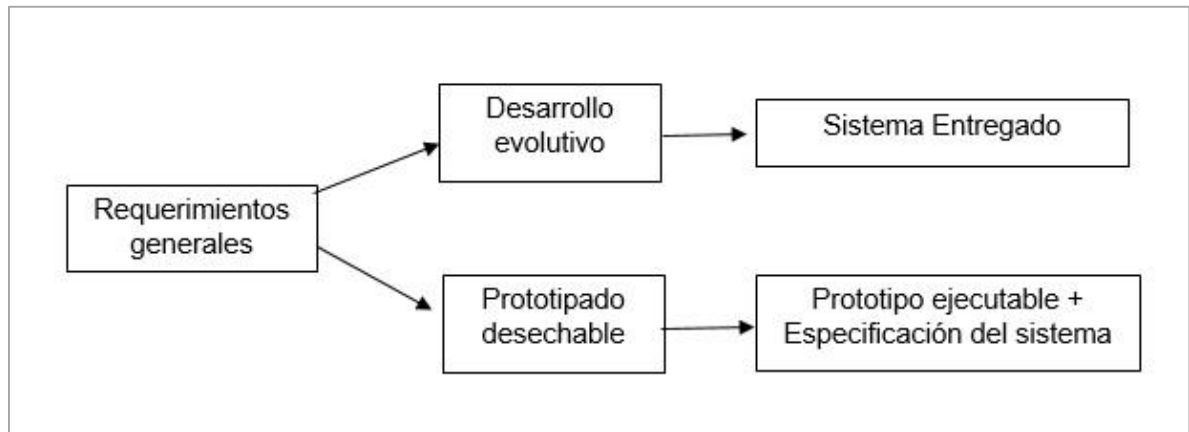


Ilustración 8 Requerimientos generales. Fuente [5]

El descontento con estos enfoques pesados condujo a varios desarrolladores de software en los años 90 a proponer nuevos métodos de desarrollo ágiles. Estos permitieron a los equipos de desarrollo centrarse en el software mismo en vez de su diseño y documentación. 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 requerimientos del sistema normalmente cambiaban rápidamente durante el proceso de desarrollo. Ha continuación se mostrara un tabla con los principios fundamentales para las metodologías ágiles.

Principio	Descripción
Participación del cliente	Los clientes deben estar fuertemente implicados en todo el proceso del desarrollo. su papel es proporcionar y priorizar nuevos requerimientos del sistema y evaluar las iteraciones del sistema
Entrega incremental	El software se desarrolla en incrementos, donde el cliente especifica los requerimientos a incluir en cada incremento
Personas no procesos	Se deben reconocer y explotar las habilidades del equipo de desarrollo. Se les debe dejar desarrollar sus propias formas de trabajar.
Aceptar el cambio	Se debe contar con que los requerimientos cambian, por lo que el sistema se diseña para dar cabida a estos cambios.

Mantener la simplicidad	Se deben centrar en la simplicidad tanto en el software a desarrollar como en el proceso de desarrollo. Donde sea posible, se trabaja activamente para eliminar la complejidad del sistema.
-------------------------	---

Tabla 2 Principios del agilísimo [15]

El manifiesto ágil en [30] define una serie de principios y valores que direccionan al equipo de proyecto de software. S. Rivadeneira en [31] manifiesta que generalmente no se cuenta con especificaciones concretas relacionadas con aspectos metodológicos.

Donde se puede obtener esa guía concreta es con estas cuatro metodologías (XP, Scrum, DSDM6 y FDD7). Estas se destacan por su especificación respecto de los procesos, practicas, roles y productos Pero en lo que respecta a las etapas de inicio del proyecto y especificación de requerimientos XP Y Scrum son las que proporcionan procesos, practicas, actividades y artefactos específicos así como la gestión del proyecto. El ciclo de desarrollo dentro de las metodologías ágiles es iterativo e incremental, esto permite entregar el software en pequeñas partes llamado entregables. S. Rivadeneira en [31] afirma que cada iteración se puede definir como un sub-proyecto en el que las actividades de análisis de requerimientos, diseño, implementación y testing se llevan a cabo con el objetivo de producir una parte del trabajo final. Este proceso se repite varias veces produciendo un nuevo entregable en cada ciclo hasta que se tenga el producto completo. Aunque todas las metodologías ágiles adoptan este ciclo, cada una presenta sus características. Entre las metodologías que cubren las primeras etapas dentro de un proyecto de software se puede especificar la Programación Extrema (XP) y Scrum.

3.1.1 Programación Extrema

Jhon Brewer en [32] afirma que en la programación extrema todos los requerimientos se expresan como escenarios llamados Historias de usuario. Las cuales se implementan directamente como una serie de tarea. Los desarrolladores trabajan en parejas y desarrollan pruebas para cada tarea antes de escribir el código. Todas las pruebas se deben de ejecutar satisfactoriamente cuando el código nuevo se integre al sistema. En los procesos de XP, los clientes están

⁶ Dynamic System Development Methods

⁷ Feature-Driven Development

fuertemente implicados en la especificación y establecimiento de prioridades de los requerimientos del sistema. Los requerimientos no se especifican como una lista de funciones requeridas del sistema. Más bien los clientes del sistema son parte del equipo de desarrollo y discuten escenarios con otros miembros del equipo. El objetivo principal de XP es de satisfacer por completo la necesidad del cliente, por eso este es integrado como parte del equipo de desarrollo.

Fue creada con el propósito de trabajar en proyectos donde el cliente todavía no le ha dado prioridad a sus necesidades, esto hace que hayan cambios constantes en los requisitos que debe cumplir la aplicación, por eso XP se adapta a las necesidades del cliente y por este motivo la aplicación se va reevaluando en periodos cortos de tiempo.

XP está pensada para el desarrollo de aplicaciones que requieran de un grupo pequeño de desarrolladores, Porque esta metodología le da mucha **importancia a la comunicación** ya que por medio de ella se tiene más claro lo que se necesita para el desarrollo de una aplicación eficiente. La comunicación debe realizarse entre programadores, jefes de proyecto y clientes.

3.1.1.1 Fundamentos de Extreme Programming

Las metodologías tradicionales presentan una serie de ambigüedades en el momento de abordar un proyecto. P. Letelier & M. Penades en [33] aseguran que XP se apoya en 3 elementos que permiten que un proyecto de desarrollo no tenga problemas en el futuro y en el momento de establecer los requisitos del proyecto sea una fase fácil de lograr. La comunicación es el primer paso para que los programadores y el cliente estén en constante comunicación para satisfacer requisitos y así tomar una fotografía exacta de lo que necesita el cliente, siempre en esta toma de requisitos se busca hacerlo de una forma detallada y precisa aunque en muchas ocasiones se consigue una lista muy extensa, esto se da porque a veces el cliente no sabe que es lo que necesita en realidad para cubrir sus necesidades, por eso las necesidades y prioridades varían durante la vida del proyecto, estos cambios son protegidos con un mecanismo de control establecido. Por otro lado P. Letelier & M. Penades en [33] declaran que la simplicidad en el desarrollo este ayuda a reducir los errores dentro de las funcionalidades que tenga el software y además permite que se adapte a los cambios que tenga el aplicativo en el futuro ya que cuando se requiera extender el proyecto no se estará obligado a descartar lo que se lleva por causa de un desarrollo complejo o enredado. Por ultimo pero no menos importante esta la realimentación dentro el proyecto esta parte le ofrece al cliente la posibilidad de tener un sistema de acuerdo a sus verdaderas necesidades ya que estas surgen a medida que se va trabajando, esta realimentación se hace con tiempo por eso tiene cabida dentro de las etapas de desarrollo con el fin de poder hacer el cambio sin afectar los costos del proyecto.

3.1.1.2 Modelo de requerimientos y Extreme Programming

Este proyecto tiene como centro la gestión de requisitos enfatizándose en las actividades que se realizan dentro de esta etapa y para esto se busca analizar que practicas agiles y artefactos suelen aplicarse en las metodologías antes mencionadas. En XP como ya se había mencionado anteriormente la comunicación es de suma importancia para la toma de requisitos como para el diseño. Entre los artefactos que tiene Extreme Programming para la toma de requisitos se tienen las historias de usuarios, Tareas de Ingeniería y tarjetas CRC.

3.1.1.2.1 Historias de usuario

D. Wells & P. Letelier en [33][34] afirman que esta es la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe tener; sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento estas pueden romperse o pueden ser remplazadas por otras más específicas, añadirse nuevas o ser modificadas. Cada Historia de usuario es lo suficiente comprensible y delimitada para que los programadores puedan implementar en cada semana.

Historia de Usuario	
Número:	Nombre Historia de Usuario:
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción:	
Observaciones:	

Tabla 3 Historias de Usuario Fuente [35]

Las Historias de Usuario tienen tres aspectos:

- Tarjeta: en ella se almacena suficiente información para identificar y detallar la historia.
- Conversación: cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación)
- Pruebas de Aceptación: permite confirmar que la historia ha sido implementada correctamente.

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro. y Nombre):
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

Tabla 4 Caso de prueba aceptación[35]

3.1.1.2.2 Task Card (Tareas De Ingeniería).

Estas se usan para describir las tareas que se realizan dentro del proyecto, el instituto Abaco en [35] expresa que las tareas pueden ser: desarrollo, corrección, mejora, entre otras. Estas tareas tienen relación con una historia de usuario; se especifica la fecha de inicio y fin de la tarea, se nombra al programador responsable de cumplirla y se describe el objetivo que se busca cumplir al realizar esta tarea.

Tarea de Ingeniería	
Número Tarea:	Historia de Usuario (Nro. y Nombre):
Nombre Tarea:	
Tipo de Tarea : Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

Tabla 5 Tarea de Ingeniería [35]

3.1.1.2.3 Tarjetas CRC (Clase - Responsabilidad – Colaborador).

CRC propone método de trabajo, preferentemente grupal, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y cómo colaboran con otros para realizar tareas. En estas tarjetas se registran los nombres de las clases, sus responsabilidades y la relación que tiene con otras clases; cada una de estas representa una clase dentro del sistema. Este proceso consta de dos etapas: la lluvia de ideas y el juego de rol. Es durante este proceso que se investiga una solución orientada a objetos del sistema, permitiendo la creatividad mediante la iteración del grupo de trabajo.

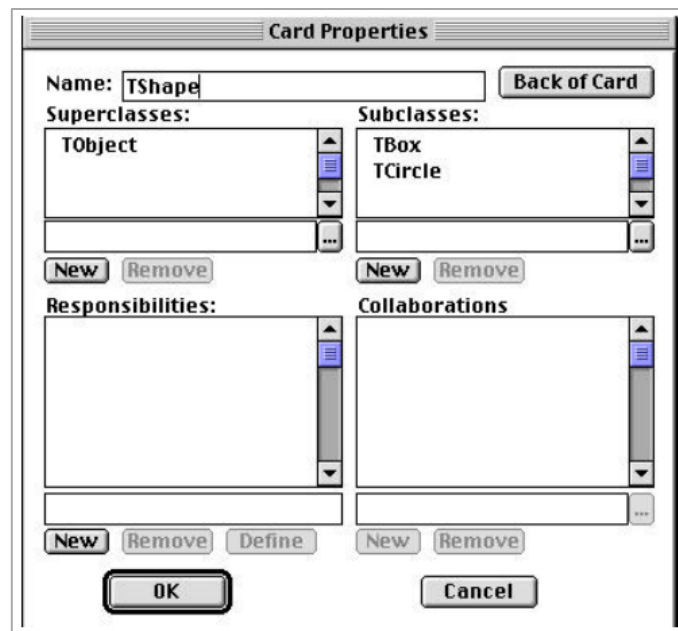


Ilustración 9 Tarjeta de CRC [36]

Lluvia de ideas - seleccionando las clases.

La primera etapa dentro del proceso sirve para identificar las clases que conformaran parte del sistema. Estas etapas son realizadas en grupo, se aconseja que los grupos sean entre 5 y 6 personas para que haya mayor flujo de ideas

Juego de rol colaboraciones y responsabilidades.

Vallespir en [37] define que la idea del juego de rol es hacer una representación teatral del sistema. Para esto se reparten las tarjetas CRC entre los integrantes del equipo (los "actores de la obra"). Cada integrante va a personificar, tal cual un actor de teatro, una o varias clases (tarjetas CRC que le tocaron al momento de repartirlas). La "obra", o más bien el guion de la "obra", es un escenario del sistema.

La idea es hacer una “obra” para cada escenario, de esta manera se hacen actuaciones de todo el sistema.

Esta actividad es interactiva y dinámica, y a través de ella se logra un acoplamiento Total del equipo de trabajo produciendo e incentivando el pensamiento creativo.

3.1.2 Scrum

J. Palacio en [38] define que Scrum es una metodología de desarrollo ágil, que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción. Aunque surgió como modelo para el desarrollo de productos tecnológicos también se emplean en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; Situaciones que se presentan con frecuencia en el desarrollo de sistemas de softwares. Fedesoft, y el Servicio Nacional de Aprendizaje (SENA) en Colombia [39],[40] capacitaron a 1.158 profesionales de TI en la implementación del modelo Scrum (metodología ágil de desarrollo de software) en el año 2010 ya que La iniciativa permitirá a las compañías de TI y desarrolladores de software mejorar el nivel de productividad, lograr altos niveles de calidad y administrar eficientemente los proyectos en las empresas en términos de tiempo y costo, consigna la información. Fedesoft & Bnamerica en [41],[42] afirman que muchas empresas en Colombia han acogido esta metodología ágil para mejorar sus procesos de producción. P. Hundermark, P. Juan & R. Claudia en [43],[44],[45] enuncian que Scrum presenta las siguientes características: Adopta estrategias de desarrollo incremental en lugar de planificación y ejecución completa del producto y como uno de sus principios tiene establecido la calidad del resultado de sus productos. Esta metodología está fundamentada en el trabajo en equipo se puede adoptar de una forma técnica, aplicando reglas definidas o pragmáticas.

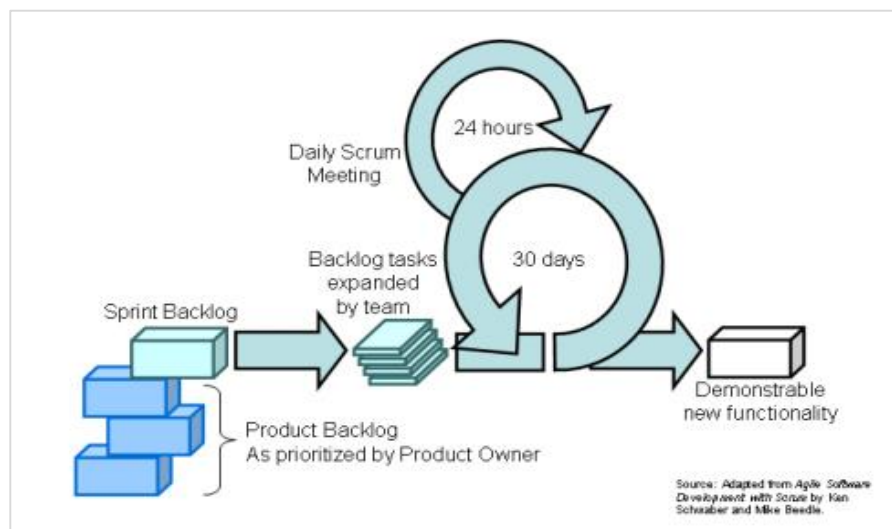


Ilustración 10 Esquema General Scrum [46]

3.1.2.1 Modelo de requerimientos y Scrum

S. Rivadeneira & G. Vilanova en [31] afirman que Scrum no se ocupa de los detalles relacionados con el modelado de requerimientos, pero claramente el artefacto Product Backlog recoge los requerimientos del sistema y sus estimaciones. Los desarrolladores no pueden implementar los requerimientos si no están en el Product Backlog. En cada iteración, el equipo de desarrollo comienza con la lista priorizada de requerimientos almacenada en el Sprint Backlog. Luego, cuando los requerimientos surgen y evolucionan debido al entorno dinámico del negocio o una solicitud de cambio de las partes interesadas, la modificación de las prioridades de los requerimientos en el product backlog se produce sobre la base actual del valor de los requerimientos de negocio. Scrum define solo 4 artefactos pero en este proyecto se tomarán 2 porque son los directamente involucrados en ese primer proceso que es la toma de requisitos Product Backlog y Sprint Backlog.

3.1.2.2 Product Backlog

N. Gandelman en [46] menciona que el Product Backlog es una lista priorizada que define el trabajo que se va a llevar a cabo en el proyecto. Cualquiera puede agregar ítems al Backlog, pero sólo el Product Owner tiene el derecho de determinar el orden en el que serán desarrollados. Cuando un proyecto comienza es muy complejo tener claro los requerimientos sobre el producto, sin embargo se determinan los más importantes que son los indicados para un Sprint. P. Hundermark en [44] declara que los requisitos son inestables en muchas de las ocasiones, Es por eso que el Product Backlog es un documento vivo, que requiere una constante preparación para mantenerlo actualizado y útil con el paso del tiempo.

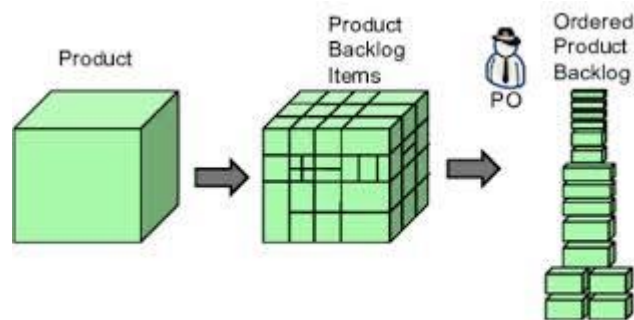


Ilustración 11 Product Backlog Items[47]

3.1.2.3 Sprint Backlog

Este es el tablero de tareas, que es simplemente una representación física al trabajo al que se ha comprometido el equipo para lo que resta del Sprint. El tablero de tareas es una señal visual por que comunica al equipo o a cualquiera que desee saber que se ha planificado y el estado actual de las actividades. Eclipse.org en [48] declara que el sprint Backlog se mantiene comúnmente como una hoja de cálculo de Excel, pero también es posible utilizar el sistema de seguimiento de defectos o cualquiera de una serie de productos de software diseñados específicamente para Scrum o ágil. Un ejemplo de la Pila de Sprint en Excel se ve así:

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Ilustración 12 Sprint Backlog[48]

3.2 Beneficios de Scrum según la industria

3.2.1 Ceiba Software House SAS

- Gestión regular de las expectativas del cliente
- Victorias tempranas dado las entregas frecuentes del producto.
- Flexibilidad y adaptación, que permiten cambios en cualquier momento adaptándose a la naturaleza de los negocios actuales y la del software con ente vivo y cambiante.
- Se trabaja de principio a fin bajo el concepto de realizar lo que más valor genera para el negocio
- Se obtiene un retorno de Inversión (ROI) más rápido dado que se realiza lo mínimo necesario para operar y posteriormente se refina el desarrollo.
- La rápida visualización que tiene el cliente sobre el producto o parte del producto evita que se lleven sorpresas al final bajando el costo por errores encontrados.
- Acceso a tarifas más económicas.
- Se aprovecha de mejor forma la sinergia de equipo.

3.2.2 Softmanagement S.A

Softmanagement en [49] define que esta metodología permite producir entregas funcionales de los sistemas, en periodos muy cortos de tiempo (SPRINTS de 2 a 4 semanas), agilizando los procesos de pruebas e identificación de incidencias, y permitiendo que los clientes puedan hacer uso del sistema y familiarizarse con sus funcionalidades, desde etapas tempranas del proyecto.

Uno de los objetivos o características fundamentales de esta metodología es que los equipos de trabajo pueden responder ágilmente a los cambios generados sobre los procesos o necesidades del negocio. El Product Owner, que también podría ser visto como el Gerente del Proyecto, tiene la potestad de cambiar las prioridades de los ítems que van a ser construidos dentro de cada una de las iteraciones (SPRINTS) definidas en el proyecto, y de esa forma estructurar el desarrollo de cualquier cambio o modificación, sin tener que impactar la totalidad del diseño o la implementación, de un sistema complejo.

CAPITULO 4

Este capítulo tiene como objetivo, dar a conocer de forma detallada los resultados que se obtuvieron durante el proceso de investigación a través de la encuesta realizada a Empresas dedicadas al desarrollo de software para el sector salud.

4. Resultados

4.1 Metodologías utilizadas para el desarrollo Ágil de software

En esta sección se presentan los resultados obtenidos de la encuesta realizada a empresas dedicadas al desarrollo de software para el sector salud en Medellín. Esta encuesta fue realizada desde el mes de abril hasta el mes de junio inclusive del año 2015. Un total de 10 empresas pudieron contestar las encuestas, Todas trabajan con metodologías de desarrollo ágil. Como metodología más utilizada se tiene SCRUM con un 90%, seguida de Programación Extrema (XP) con un 60% y Por última a Kanba con un 50%

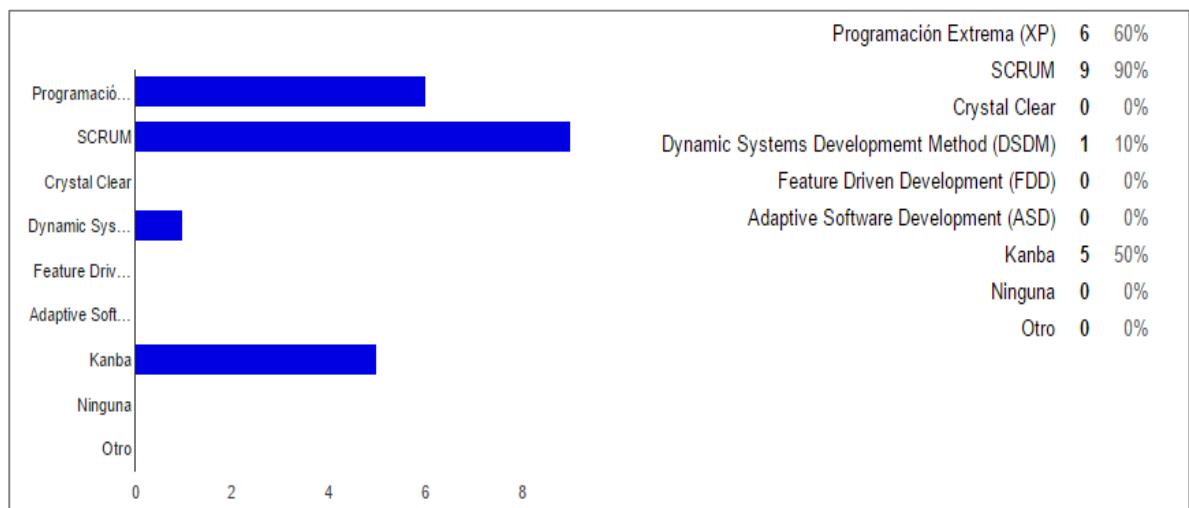


Ilustración 13 Metodologías ágiles Fuente: Propia Autoría

4.2 Técnicas utilizadas para la Elicitación de requisitos

En la Ilustración 14 se observa que las técnicas menos utilizadas en las empresas son desarrollo conjunto de Aplicaciones (JAD) y estudios de documentación y a la misma vez las técnicas más utilizadas por los encuestados son: Entrevista, Encuesta, Lluvia de ideas y desarrollo de prototipos. (Entrevista: 90%, Encuesta: 80%, Lluvia de ideas: 70%, Desarrollo de prototipos: 70%)

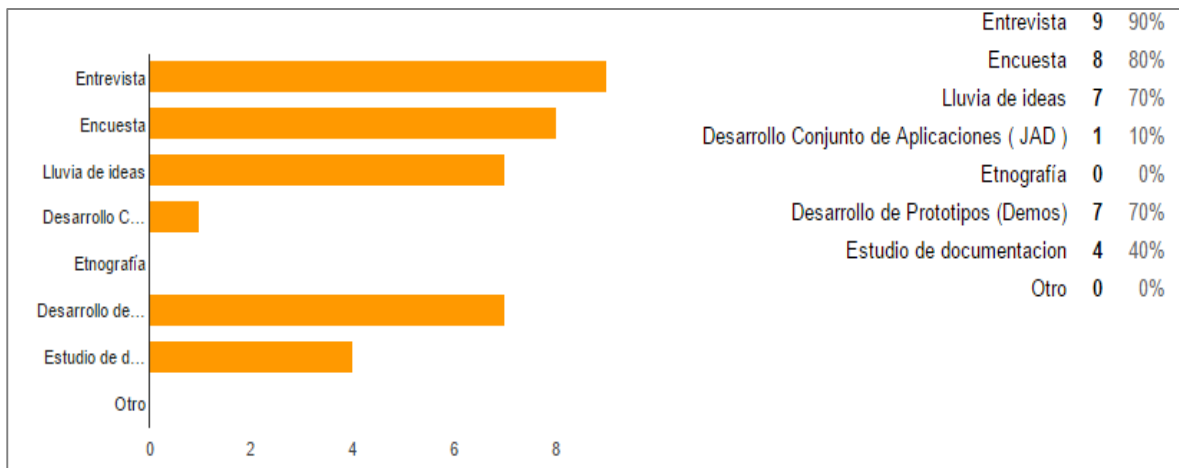


Ilustración 14 Técnicas para la Elicitacion de Requisitos Fuente: Propia Autoría

4.3 Obstáculos para la Adopción Ágil

En la Ilustración 15 se puede observar alguno de los obstáculos que presentaron las empresas en el momento de adoptar las metodologías ágiles como método de trabajo para el desarrollo de software. Uno de los mayores obstáculos que se tiene a la hora de adoptar métodos de trabajo ágil es la complejidad del proyecto y colaboración con el cliente con un 70% seguido del cambio de la cultura organizacional y resistencia general al cambio con un porcentaje del 50% ya que las personas están acostumbradas a llevar a cabo sus actividades de una forma distinta en la cual ya tiene la experiencia de cómo hacerlo. Por otra parte el 40% se debe a La disponibilidad de personal con conocimientos adecuados, Mientras que el apoyo a la gestión y la colaboración con el cliente tiene un porcentaje del 20%.

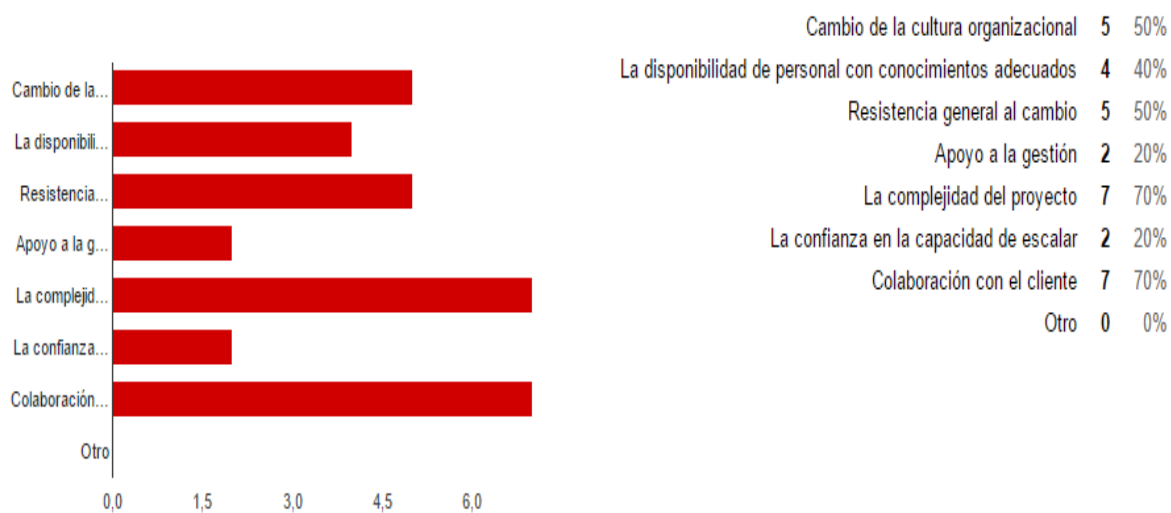


Ilustración 15 Obstáculos para la adopción Ágil Fuente: Propia Autoría

4.4 Beneficios que se obtendrían al aplicar de forma adecuada metodologías ágiles en proyectos de desarrollo.

Para desarrollar un software de calidad que tenga un rendimiento eficiente, que sea fácil de manejar y que ahorre recursos es necesario tener presente la ingeniería de requisitos y las metodologías ágiles ya que estas brindan calidad y mejoramiento en los procesos dentro del desarrollo. En la Ilustración 16 se encuentran los beneficios que se obtendrían al aplicar de forma adecuadas las metodologías ágiles. Para las empresas encuestas el mayor beneficio es la reducción en los costos con un porcentaje del 80%, seguido del Aumento de la productividad y la rápida respuesta a cambios de los requisitos con un 70%, la calidad mejorada y la mejora a la gestión de riesgo posee un 60% y por último la simplificación de la sobrecarga de procesos con un 50%.

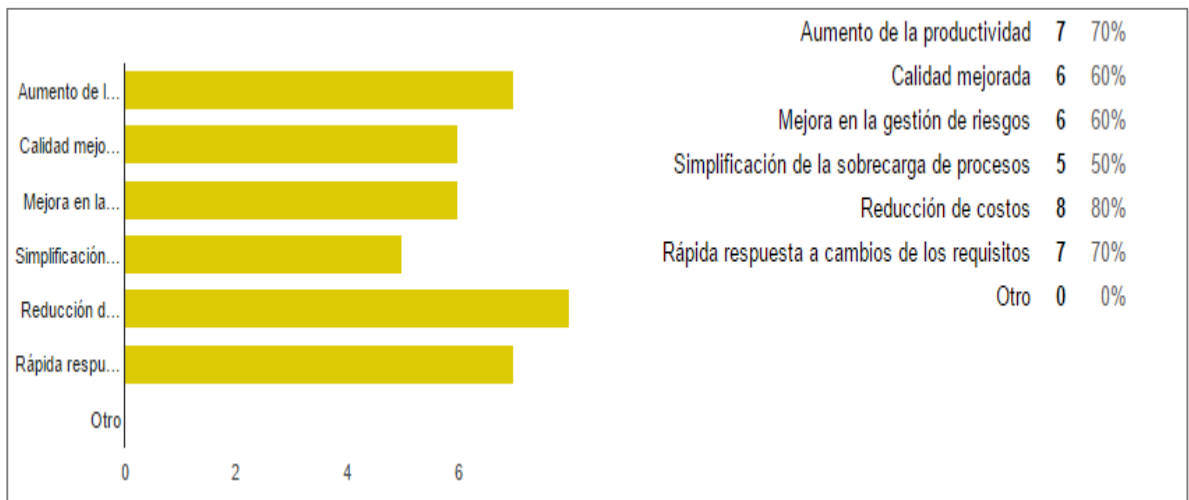


Ilustración 16 Beneficios aplicación de metodologías Ágiles Fuente: Propia Autoría

4.5 Técnicas usadas para la priorización de Historias de usuario.

Las técnicas de priorización son usadas para determinar los requisitos más valiosos para el cliente. Como técnica más utilizada por las empresas para llevar a cabo este proceso está la Matriz de Priorización con un 100% de aceptación seguida de la técnica Theme Scoring con un 50%, Análisis de Kano 40%, y Moscow con un 30%

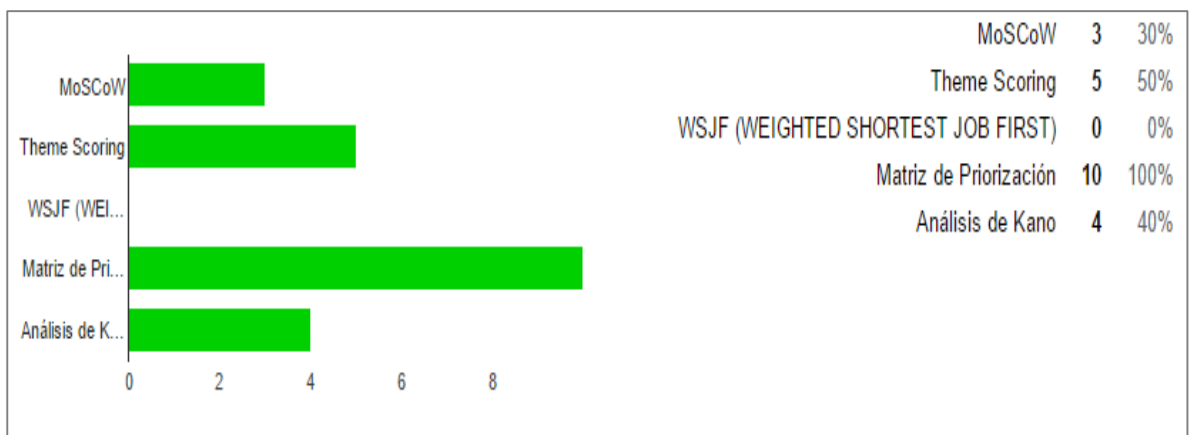


Ilustración 17 Técnicas de Priorización Fuente: Propia Autoría

¿En su empresa obtienen información sobre el dominio del problema y el sistema actual en el proceso de elicitation de requisitos?

En la Ilustración número 18 se observa que un 50% de las empresas encuestas siempre obtienen información sobre el dominio del problema y que el 40% lo hace con frecuencia mientras que el 10% lo hace rara vez.

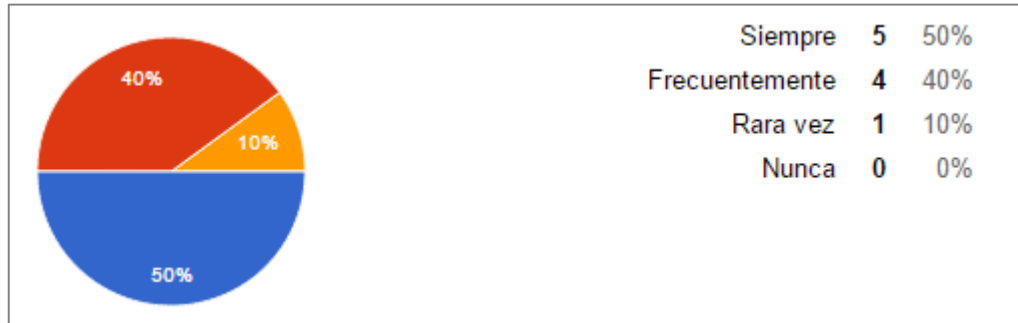


Ilustración 18 Información proceso de elicitation Requisitos Fuente: Propia Autoría

¿En su empresa preparan y realizan las sesiones de elicitation y negociación de requisitos para realizar la visita inicial al usuario y/o cliente del sistema?

En la Ilustración número 19 se observa que un 40% de las empresas encuestas preparan y realizan las sesiones de elicitation y negociación de requisitos para realizar la visita inicial al usuario y/o cliente del sistema y que el 60% lo hace con frecuencia.



Ilustración 19 Preparación de sesiones de Elicitacion Fuente: Propia Autoría

En su empresa, después de la obtención del dominio del problema y el sistema actual y después de la sesión de elicitation y negociación de requisitos. ¿Realizan identificación y revisión de los objetivos del sistema?

En la Ilustración número 20 se observa que un 60% de las empresas encuestas siempre realizan identificación y revisión de los objetivos del sistema después de la obtención del dominio del problema y el sistema actual y después de la sesión de elicitation y negociación de requisitos y que el 40% lo hace con frecuencia, mientras que el 10% lo hace rara vez.

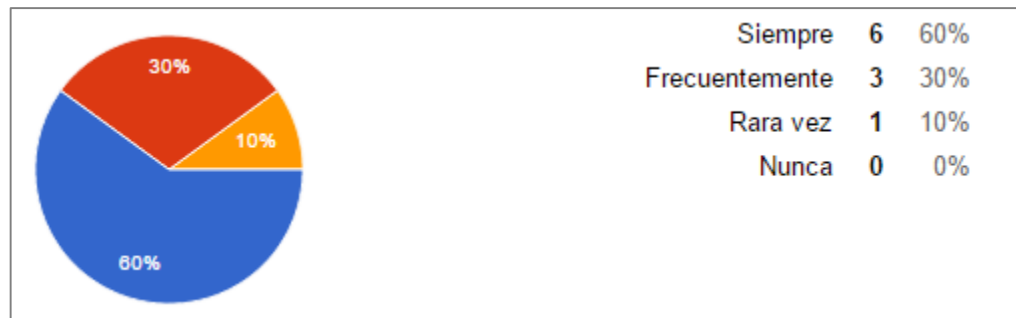


Ilustración 20 Identificación y revisión de los objetivos del sistema Fuente: Propia Autoría

¿En su empresa revisan los requisitos de información presentados por el usuario?

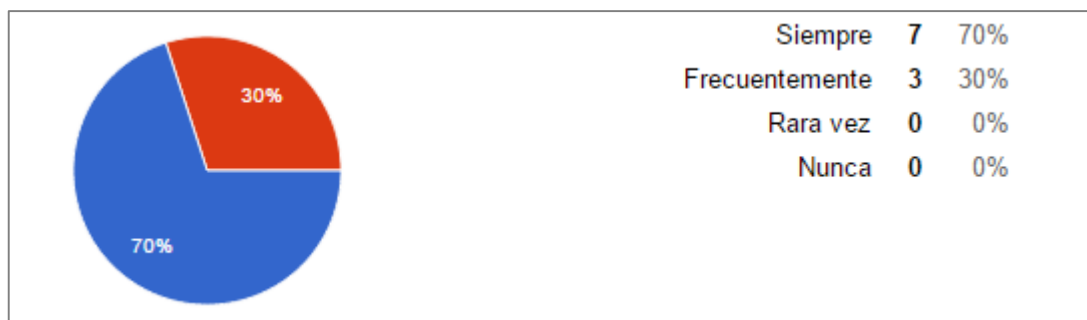


Ilustración 21 Requisitos de Información presentadas por el usuario Fuente: Propia Autoría

¿En su empresa revisan los requisitos funcionales presentados por el usuario para una posible retroalimentación?

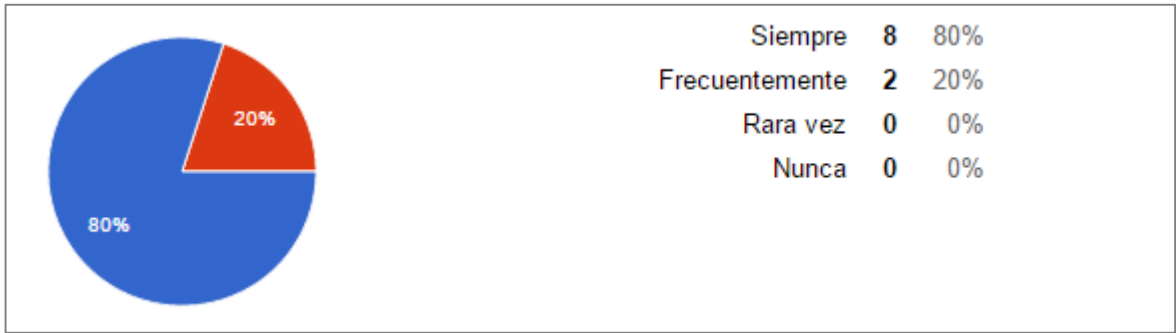


Ilustración 22 Revisión requisitos funcionales por el usuario Fuente: Propia Autoría

¿En su empresa revisan los requisitos no funcionales presentados por el usuario para una posible retroalimentación?

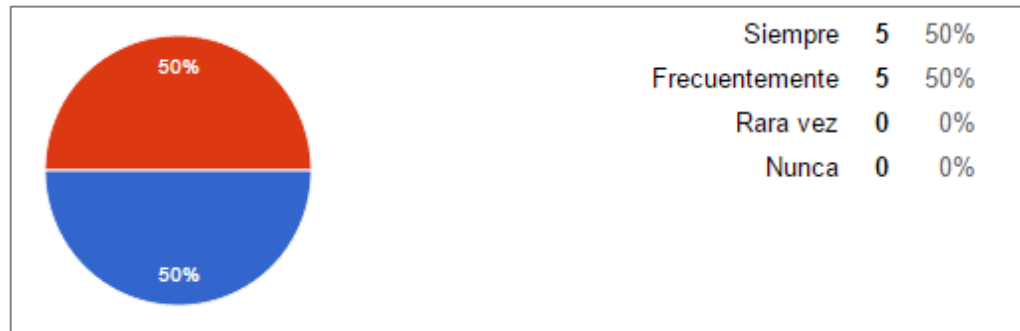


Ilustración 23 Revisión requisitos no funcionales Fuente: Propia Autoría

¿Priorizan los objetivos y requisitos para comenzar las actividades de análisis, diseño y desarrollo del software?



Ilustración 24 Priorización de los objetivos y requisitos Fuente: Propia Autoría

¿En su empresa una vez priorizados los requisitos realizan historias de usuarios y crean el product Backlog?

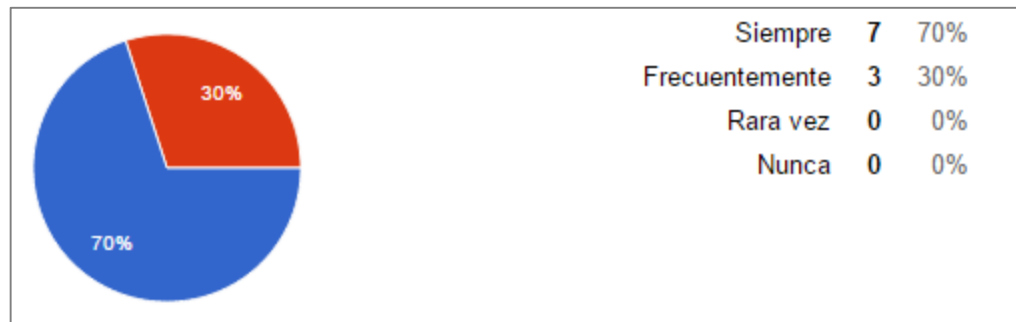


Ilustración 25 Realización de historias de Usuarios y Product Backlog Fuente: Propia Autoría

4.6 Empresas encuestadas dedicadas al desarrollo de software para el sector salud en Medellín.

Metodologías Ágiles	
SCRUM	
Programación Extrema (XP)	
Kanba	
Dynamic Systems Development Methods (DSDM)	

Nombre de la empresa encuestada	Metodologías para el desarrollo Ágil				NIVEL CMMI
Aplisalud	XP	SCRUM			
MVM Ingeniería de Software	XP	SCRUM	Kanba		V
ilimitada S.A.S.	XP	SCRUM	Kanba		III
Carvajal		SCRUM			III
Deconsoft		SCRUM			
VICSOFT S.A.S		SCRUM	Kanba		
Ceere	XP	SCRUM	Kanba		
Geminus S.A.S	XP	SCRUM	Kanba		
Sicme	XP			DSDM	
DigitalWare		SCRUM			

Tabla 6 Empresas Encuestadas Fuente: Propia Autoría

4.7 Encuesta ingeniería de requisitos empresas dedicadas al desarrollo de software en el sector de la salud en Medellín.

***Obligatorio**

Nombre de la empresa encuestada

¿Que metodologías utilizan para el desarrollo ágil de software? *

- Programación Extrema (XP)
- SCRUM
- Crystal Clear
- Dynamic Systems Development Method (DSDM)
- Feature Driven Development (FDD)
- Adaptive Software Development (ASD)
- Otros
- Kanba
- Otro:

¿Que técnicas utilizan para la elicacion de requisitos? *

- Entrevista
- Encuesta
- Lluvia de ideas
- Desarrollo Conjunto de Aplicaciones (JAD)
- Etnografía
- Desarrollo de Prototipos (Demos)
- Estudio de documentacion
- Otro:

Ilustración 26 Encuesta Empresas Pt 1 Fuente: Propia Autoría

¿Cuales han sido los mayores obstáculos para la adopción Ágil? *

- Cambio de la cultura organizacional
- La disponibilidad de personal con conocimientos adecuados
- Resistencia general al cambio
- Apoyo a la gestión
- La complejidad del proyecto
- La confianza en la capacidad de escalar
- Colaboración con el cliente
- Otro:

¿Qué beneficios se obtendrían al aplicar de forma adecuada metodologías ágiles en proyectos de desarrollo? *

- Aumento de la productividad
- Calidad mejorada
- Mejora en la gestión de riesgos
- Simplificación de la sobrecarga de procesos
- Reducción de costos
- Rápida respuesta a cambios de los requisitos
- Otro:

¿En su empresa obtienen información sobre el dominio del problema y el sistema actual en el proceso de elicitation de requisitos? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

Ilustración 27 Encuesta empresas Pt2 Fuente: Propia Autoría

¿En su empresa preparan y realizan las sesiones de elicitation y negociacion de requisitos para realizar la visita inicial al usuario y/o cliente del sistema? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

¿En su empresa, después de la obtención del dominio del problema y el sistema actual y después de la sesión de elicitation y negociación de requisitos. Realizan identificación y revisión de los objetivos del sistema? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

¿En su empresa revisan los requisitos de información presentados por el usuario? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

¿En su empresa revisan los requisitos funcionales presentados por el usuario para una posible retroalimentacion? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

Ilustración 28 Encuesta empresas Pt3 Fuente: Propia Autoría

¿En su empresa revisan los requisitos no funcionales presentados por el usuario para una posible retroalimentación? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

¿Priorizan los objetivos y requisitos para comenzar las actividades de análisis, diseño y desarrollo del software? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

¿En su empresa una vez priorizados los requisitos realizan historias de usuarios y crean el product backlog? *

- Siempre
- Frecuentemente
- Rara vez
- Nunca

¿En cada sprint que técnicas usan en su empresa para priorizar las historias de usuario? *

- MoSCoW
- Theme Scoring
- WSJF (WEIGHTED SHORTEST JOB FIRST)
- Matriz de Priorización
- Análisis de Kano

Ilustración 29 Encuesta Empresas Pt4 Fuente: Propia Autoría

4.8 Conclusiones

El análisis realizado en este proyecto está enfocado al proceso de elicitación de requisitos en las metodologías ágiles; Por medio de una encuesta realizada a una muestra de 10 empresas que se dedican a desarrollar software para el sector de la salud en Medellín, se busca mostrar cuales son las metodología más utilizada dentro de las empresas encuestadas y los beneficios y obstáculos que han tenido que afrontar para la adopción de metodologías ágiles. De las empresas encuestadas como metodología que más utilizan para el desarrollo de software es SCRUM con un proporción del 90%; esta metodología es elegida por sus principios y su esquema de trabajo sencillo pero con un estricto control y seguimiento durante cada etapa, Además se realizó un estado del arte referente a la ingeniería de requisitos y los métodos ágiles. Ya que por medio de una buena implementación de los mismos se logra el éxito en un proyecto de software. La industria de software en Colombia están adoptando metodologías ágiles que le permite agilizar la toma de requisitos para poder adaptarlas en proyectos de alto rendimiento y que se pueden obtener parte del mismo desde las primeras semanas de desarrollo por medio de los entregables o sprint que tienen un límite de tiempo de 3 a 4 semanas. Es muy importante indicar que el poder formular una especificación de requerimientos completa y consistente, es un paso muy importante para evitar cometer errores en la definición de los mismos, por qué una vez finalizado el sistema sería muy costoso hacer cambios. De ahí, la vital importancia que tiene la ingeniería de requisitos en generar una adecuada especificación que atienda claramente y sin ambigüedades los requerimientos del sistema a desarrollar, con el fin primordial de evitar que los proyectos fracasen debido a una mala elaboración de la definición y especificación de requerimientos.

Los principales aportes de este trabajo son:

- Con esta investigación se ha podido mostrar como la ingeniería de requisitos y las metodologías ágiles pueden hacer que un proyecto de desarrollo de software pueda satisfacer las necesidades de un sistema.
- Las metodologías ágiles utilizan los artefactos para tomar los requisitos en los proyectos de desarrollo de software.
- SCRUM como la metodología de trabajo más aceptada por las empresas encuestadas.
- Ingeniería de requisitos como el principio para desarrollar software de calidad en conjunto con las metodologías ágiles para ahorrar costos, agilizar el trabajo y poder hacer entregas al cliente desde las 4 primeras semanas.

- Las técnicas más utilizadas para la toma de requisitos son la entrevista, la encuesta y la lluvia de Ideas
- Los beneficios que se obtienen al aplicar metodologías ágiles son la reducción de costos, la calidad mejorada y aumento de la productividad.
- Los mayores obstáculos que han tenido estas empresa para la adopción de metodologías ágiles ha sido el cambio de la cultura organizacional, la colaboración con el cliente y la disponibilidad de personal con los conocimientos adecuados en algunos proyectos.

4.9 Trabajos futuros

- Investigar cuales son los procesos y retos que las empresas de desarrollo de software tienen que pasar para lograr la certificación en CMMI.
- saber que implica estar en cada uno de sus niveles
- Mostrar si en realidad el modelo CMMI garantiza el éxito del proceso de mejora en la organización.
- Investigar que empresas están certificadas en CMMI en Medellin y a qué nivel corresponden.

4.10 Referencias Bibliográficas

- [1] R. S. Pressman, *Ingeniería del software. Un enfoque práctico*. 2002.
- [2] ALONSO TORO LAZO and L. C. BENJUMEA, “ESTADO DEL ARTE DE LA INGENIERÍA DEL SOFTWARE EN EL ÁMBITO NACIONAL E INTERNACIONAL DE ACUERDO A ORGANIZACIONES QUE TRATAN LA DISCIPLINA,” Universidad católica popular de risaralda, 2010.
- [3] R. María and T. De Paz, “El proceso de Ingeniería de Requisitos en el ciclo global del software,” Universidad de Sevilla.
- [4] P. Dunne, “Software Engineering: analysis and design,” 1989.
- [5] R. Pressman, *Ingeniería del Software -Roger Pressman 6th.Ed.pdf*, Sexta Edic. 2005.
- [6] S. G. Gunda, “Requirements Engineering : Elicitation Techniques,” University West, 2008.
- [7] A. Hall, “No hay balas de plata . Esencia y accidentes de la ingeniería de software,” 2006.
- [8] N. Valeria Andriano, “Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado . Propuesta de métricas para la elicitación .,” Universidad Blas Pascal . Universidad Nacional de la plata, 2006.
- [9] E. S. M, *Libro Blanco de la Ingeniería de Software en América Latina* .
- [10] A. R. Fernandez, “CLASIFICACIONES DE TIPOS DE REQUISITOS PARA LA MEJORA DEL PROCESO DE DESARROLLO DEL SOFTWARE,” 2011.
- [11] A. Rudorfer, T. Stenzel, and G. Herold, “A business case for feature-oriented requirements engineering,” *IEEE Softw.*, vol. 29, no. 5, pp. 54–59, 2012.
- [12] D. Lowe and W. Hall, *Hypermedia and the Web: an Engineering Approach*. 1999.
- [13] C. M. University, “REQUIREMENTS ENGINEERING,” 2014, 2014. [Online]. Available: http://www.sei.cmu.edu/productlines/frame_report/req_eng.htm.
- [14] M. Manies and U. Nikual, “Software Requirements Elicitation in the Context of a Software,” a University of Technology, Finland, 2011.
- [15] M. Aoyama, “Requirements Engineering Based on REBOK (Requirements Engineering Body Of Knowledge) and Its Practice,” pp. 79–80, 2013.

- [16] L. Westfall, "Software Requirements Engineering: What, Why, Who, When, and How By Linda Westfall," *ASQs Softw. Qual. Prof. J.*, no. 2004, pp. 9–15, 2006.
- [17] A. D. Toro, B. B. Jiménez, and I. Técnico, "Metodología para la Elicitación de Requisitos de Sistemas Software," Universidad de Sevilla, 2002.
- [18] I. D. E. Requerimientos, "2. Ingeniería De Requerimientos.," pp. 7–24.
- [19] V. D. D. R. I. (Universidad P. De Navarra, "Tipos de encuestas y diseños de investigación," *Catálogo Publicaciones la Univ. Púbrica Navarra*, p. 243, 2002.
- [20] Sommerville, *Ingeniería del Software*, Séptima. Madrid España, 205AD.
- [21] I. Sommerville, *Ingeniería del software*. 2005.
- [22] M. Maria, M. Sandoval, H. C. Rica, M. Maria, A. García, and H. C. Rica, "La Trazabilidad en el Proceso de requerimientos de software," 2004.
- [23] M. D. C. Gómez Fuentes, *Material Didáctico Notas del Curso Análisis de Requerimientos*. 2011.
- [24] informatica basica Emilio doñe, "IDENTIFICAR LOS REQUERIMIENTOS A LA HORA DEL DESARROLLO." [Online]. Available: <http://informaticabasica.org/como-identificar-los-requerimientos/>.
- [25] F. Chen, "From Architecture to Requirements: Relating Requirements and Architecture for Better Requirements Engineering," pp. 451–455, 2014.
- [26] P. Verónica and V. Alvarado, "Técnicas efectivas para la toma de requerimientos," 2012.
- [27] J. M. Drake, "Ingeniería Ingeniería Software software Verificación y Validación," pp. 1–33, 2009.
- [28] G. Verificaci and S. Preste, "Verificación y validación."
- [29] Instituto Nacional de Tecnologías de la Comunicación, "Guía práctica de gestión de requisitos," vol. 1, p. 21, 2008.
- [30] et al Beck, K., "The Agile Manifesto. Manifesto for Agile Software Development." [Online]. Available: <http://www.agilemanifesto.org>.
- [31] S. Rivadeneira, G. Vilanova, M. Miranda, and D. Cruz, "El modelado de requerimientos en las metodologías ágiles," pp. 383–387.
- [32] C. Jhon Brewer, "Programacion Extrema," 2001. [Online]. Available: <http://programacionextrema.tripod.com>. [Accessed: 22-Oct-2014].

- [33] P. Letelier and M. C. Penadés, “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP),” *Técnica Adm.*, vol. 5, no. 26, p. 17, 2006.
- [34] D. Wells, “Historias de usuario extremeprogramming,” 1999 - 2013. [Online]. Available: <http://www.extremeprogramming.org/>.
- [35] Instituto Superior Tecnológico Privado Abaco, “Artefactos en Extreme Programing (xp),” pERU, 2015.
- [36] H. Halbleib, “Software design Using CRC Cards.,” *Real Time Mag.*, no. 1, pp. 28–32, 1999.
- [37] D. Vallespir, “CRC y un Taller,” Unniversidad de la republica, 2002.
- [38] J. Palacio, “El modelo Scrum,” 2006, pp. 1–5.
- [39] Fedesoft.org, “Colombia capacitará a más de 1.000 profesionales en metodología ágil de desarrollo de software.” [Online]. Available: <http://fedesoft.org/colombia-capacitara-a-mas-de-1-000-profesionales-en-metodologia-agil-de-desarrollo-de-software/>.
- [40] bnamericas.com, “Colombia to train over 1,000 professionals.” [Online]. Available: <http://www.bnamericas.com/en/news/technology/colombia-to-train-over-1000-professionals-in-agile-software-development-framework?idioma=en>.
- [41] INTERSOFTWARE, “Ocho empresas de Intersoftware seleccionadas para el Programa Piloto de Transferencia de la Metodología SCRUM.” [Online]. Available: <http://www.intersoftware.org.co/content/ocho-empresas-de-intersoftware-seleccionadas-para-el-programa-piloto-de-transferencia-de-la->.
- [42] Psl, “Adopcion Scrum,” 2004, 2015. [Online]. Available: <http://www.psl.com.co/>.
- [43] P. Juan and R. Claudia, “Gestión de proyectos Scrum Manager,” 2011.
- [44] P. Hundermark, “Un Mejor Scrum,” in *I Can*, 2009, p. 29.
- [45] M. Eugenia and R. Izaquita, “AGILIZANDO LO ÁGIL: UN FRAMEWORK PARA LA DESARROLLO DE SOFTWARE BAJO EL MODELO CMMI EN COMPAÑÍAS QUE USAN METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE USANDO EL MODELO ACELERADO DE IMPLEMENTACIÓN (AIM) MARIA,” 2011.
- [46] N. Gandelman, “Universidad ORT Uruguay,” p. 12, 2005.
- [47] S. Sharma, “The Product Backlog Is Better Ordered Than Prioritized.” [Online]. Available: <https://www.scrumalliance.org/>.
- [48] Eclipse.org, “Artefacto: Sprint Backlog.”

[49] S. S.A, "Scrum en softmanagement S.A." [Online]. Available:
www.softmanagement.com.co.