LA INGENIERIA DE REQUISITOS EN LAS METODOLOGIAS ÁGILES:

REQUISITOS ÁGILES

JHOAO ANDRES GONZALEZ LONDOÑO

DIEGO ARMANDO ANDUQUIA RIVERA

UNIVERSIDAD EAFIT

ESCUELADE INGENIERÍAS

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

MEDELLIN

2012

LA INGENIERIA DE REQUISITOS EN LAS METODOLOGIAS ÁGILES:

REQUISITOS ÁGILES

JHOAO ANDRES GONZALEZ LONDOÑO

DIEGO ARMANDO ANDUQUIA RIVERA

Proyecto de Grado

Asesor

Alberto Antonio Restrepo Velásquez

UNIVERSIDAD EAFIT

ESCUELADE INGENIERÍAS

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

MEDELLIN

2012

Nota de aceptación:

_____

_____

_____

_____

_____

_____

_____

Firma del presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

Medellín, _____ de _____ de _____

AGRADECIMIENTOS

Este proyecto de grado consideramos no hubiese salido adelante sin el apoyo incondicional de personas que supieron acogernos y guiarnos por el mejor camino para obtener grandes resultados.

Queremos brindar un agradecimiento muy especial al docente Alberto Antonio Restrepo en representación de todo el área de docentes del departamento de Informática y Sistemas y quien con su paciencia y apoyo incondicional en todo el proceso como estudiantes de la carrera Ingeniería de Sistemas, fue quien nos enseño el camino que debíamos seguir para sacar adelante nuestros proyectos, como este que es de gran importancia para todos los entes que se encuentran en el entorno.

Finalmente brindamos nuestro más sincero agradecimiento a la Universidad EAFIT, dado que nos abrió sus puertas para permitir formarnos como grandes profesionales en esta gran carrera y tan prestigiosa Universidad.

CONTENIDO

LISTA DE FIGURAS

GLOSARIO

INGENIERIA: estudio y aplicación, por especialistas, de las diversas ramas de la tecnología.

METODOLOGIA: conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal.

REQUISITO: circunstancia o condición necesaria para algo.

SOFTWARE: conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

AGIL: ligero, pronto, expedito.

RASTREABILIDAD: es la aptitud para rastrear la historia, la aplicación o la localización de una entidad mediante indicaciones registradas.

ELICITACIÓN: es el traspaso en forma fluida de información de un programa a otro sin dificultades. Atrapar o incluir información de un programa a otro, o en su defecto de un usuario a otro en forma fluida y entendible.

VALIDACIÓN: Acción y efecto de validar.

VERIFICACIÓN: comprobación de la verdad o autenticidad de algo, también es entendida como relación de algo que se dijo o se había pronosticado.

RESUMEN

En la actualidad han venido tomando cada vez mas fuerza ciertas corrientes del desarrollo de software, que son comúnmente conocidas como Metodologías Ágiles, las cuales brindan un gran beneficio enfocado a un desarrollo más eficiente y efectivo del producto software, al igual que otras metodologías de desarrollo requieren de todo un proceso en el cual los requisitos juegan un papel preponderante para alcanzar el éxito en cada uno de los proyectos que se estén trabajando, dado que son estos los que estarán brindando una visión más clara a la hora de llevar a cabo la implementación del producto.

Es por esto que desde este punto de vista se hará un recorrido sobre el Proceso de Desarrollo de Software enfocado en la Ingeniería de Requisitos aplicada a lo que son las Metodologías Ágiles, las cuales garanticen una mejor calidad en la producción del software tanto a nivel nacional como internacional.

INTRODUCCIÓN

Este trabajo investigativo ha sido de gran importancia en nuestras aspiraciones profesionales ya que nos permite ir desde un marco general para llegar a términos específicos y entender la situación que en la actualidad se viene presentando con el uso de las nuevas metodologías del Proceso de Desarrollo Software, específicamente en las metodologías ágiles, en ellas es importante contemplar aspectos de vital importancia que conllevan a un buen desempeño de los proyectos, aunque en algunas ocasiones no son concebidos y aplicados de manera optima, tratando de realizar un menor esfuerzo que permita agilizar cada uno de los procesos que se llevan a cabo en la elaboración de productos Software.

No obstante dicha agilidad para llevar a cabo estos Procesos de Desarrollo puede desencadenar una serie de inconvenientes al no seguir de manera apropiada aspectos importantes o preponderantes como la Ingeniería de Requisitos que ayudan a entender mejor el entorno del problema y sentar las bases concretas de la solución que garanticen la disminución de los riesgos y aumenten la calidad de los productos.

Esperamos que este trabajo sea gran utilidad para las personas que deseen aplicar y ampliar sus conocimientos en este campo y si es el caso sirva de base para nuevos proyectos estudiantiles para la universidad y el público en general.

OBJETIVOS

OBJETIVO GENERAL

Brindar una mirada mediante un estudio investigativo de la Ingeniería de Requisitos para encontrar su aplicabilidad en el contexto de las Metodologías Ágiles que permitan enriquecer estas últimas y aportarle mayor calidad a los productos desarrollados mediante estas.

OBJETIVOS ESPECIFICOS

- Identificar las características que utilizan las Metodologías Ágiles en relación a la Ingeniería de Requisitos.

- Definir y proponer herramientas de Ingeniería de Requisitos que permitan un mayor desempeño orientado a la eficacia y calidad de las Metodologías Ágiles.

- Dar a conocer a las personas interesadas y que hacen uso de las metodologías de la importancia e implicación de la implementación de la Ingeniería de Requisitos en las Metodologías Ágiles.

- Aumentar nuestra capacidad cognitiva a través de los datos encontrados los cuales se coinvertirán en información valiosa para ser plasmada en presente trabajo.

ALCANCE

Alrededor de todo el marco de la Ingeniería de Software vamos encontrando algunos conceptos que son de gran importancia a la hora de realizar un desarrollo de software, entre las cuales podemos incorporar la Ingeniería de Requisitos, que con sus conocimientos y técnicas estará siendo parte transversal del desarrollo de un producto software, pero todo esto va ligado a un conjunto de procedimientos los cuales se enmarcan dentro de una Metodología que puede llegar a ser tradicional o ágil, aunque dentro de las segundas no se tenga un esquema especificado respecto a los requisitos y es por ello que muchos analistas se están introduciendo a indagar todo un modelo de cómo en las Metodologías Ágiles se puede llevar a cabo un buen trabajo con los requisitos.

Teniendo en cuenta lo descrito anteriormente en este trabajo se estará brindando un punto de vista de cómo se están interpretando o aplicando los requisitos como eje fundamental del Proceso de Desarrollo de Software en la fuerte corriente de las Metodologías Ágiles que cada vez cobran mayor importancia en la Ingeniería de Software.

JUSTIFICACION

Este trabajo se realiza con el fin de dejar un estudio solido de las herramientas que como Ingenieros de Sistemas debemos tener muy presentes a la hora de utilizar metodologías de software, más concretamente las Metodologías Ágiles que son una realidad que han llegado para quedarse tomando más fuerza día a día.

Este estudio surge a raíz de ver como hoy en día las Metodologías Ágiles están haciendo uso de la Ingeniería de Requisitos, para llevar a cabo de la mejor manera cada uno de los proyectos, pues el éxito de un proyecto esta soportado en las bases o lo que en la ingeniería civil es conocido como los cimientos, pues de la resistencia de estos dependerá si la casa puede tener o no una estabilidad y una mejor elaboración, esto es igual en la ingeniería de Software pues en las metodologías la base es cada uno de los requisitos que se fueron obteniendo pero en algunos de los casos las Metodologías Ágiles no están teniendo en cuenta una buena etapa de requisitos y es por esto el estudio realizado pues queremos dar a conocer cuales podrían ser las herramientas que permitan un mejor desarrollo de los proyectos software y obtener los mejores resultados.

1. INGENIERIA DE SOFTWARE

"Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software" (Zelkovitz, 1978).

Teniendo en cuenta la definición que da el señor Zelkovitz, se dice que la Ingeniería de Software es un proceso de abstracción que permite diseñar un modelo preciso y concreto en un entorno del desarrollo y mantenimiento de software, a partir de una realidad compleja, indefinida y subjetiva.

La Ingeniería de Software se puede definir como: "Un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software", donde se identifica tanto un proceso del ¿Q*ué debo hacer?* posteriormente el procedimiento del ¿C*ómo debo hacerlo?*

En conclusión esta Ingeniería es la ciencia encargada de definir los principios y metodologías para el buen desarrollo de un producto software que sea utilizable y de bajo costo en todas y cada una de las etapas del proceso de elaboración.

Es por esta razón que se expresa que no se puede tener conocimiento alguno de la solución a un problema si antes no se tiene conocimiento de este último.

1.1 PRINCIPIOS DE LA INGENIERÍA DE SOFTWARE

Dentro de la Ingeniería de Software se identifica que existen tres principios que son pilares fundamentales para el buen desarrollo de un producto software, estos principios se considera que deben ser transversales a todo el proceso de desarrollo, dichos principios son:

1.1.1 Calidad. Dado que este concepto debe estar presente desde el inicio del proyecto y como se expreso en el anterior párrafo debe ser transversal al mismo, esto porque ayuda a tener una prevención con respecto a los defectos y a la corrección de los mismos, además permite identificar estas causas o síntomas para eliminarlas de raíz y realizar una auditoría del trabajo, dado que se debe asegurar la calidad previniendo los defectos. Además que se le garantiza a los clientes o usuarios un producto de gran calidad.

1.1.2 Gestión. Donde esta debe definir cuáles son los roles y responsabilidades de cada uno de los involucrados en el proyecto, definir cuál es la planeación del trabajo que amerita el desarrollo,

realizar un monitoreo continuo en cuanto a la evolución de la planeación para de alguna manera tomar decisiones que les permita mejorar y refinar dicha planeación.

1.1.3 Ingeniería. Que es a través de la cual se debe definir e identificar los problemas, en caso de ser complejos o muy grandes realizar una división del mismo en pequeños sub-problemas pero teniendo en cuenta cual es la relación que tienen cada uno de ellos entre sí, todo esto utilizando el conjunto de conocimientos y técnicas que han surgido y que son de gran aplicabilidad a lo que involucre la ingeniería.

La ingeniería de Software es de gran ayuda para las personas que trabajan en el desarrollo, dado que les muestra unos lineamientos o metodologías que deben ser utilizados en la correcta elaboración de productos software, que al final serán entregados directamente al cliente y usuario del mismo.

Son todos estos lineamientos o metodologías las que les van a mostrar cual es el camino más adecuado que se debe tomar para obtener al final un excelente producto, lineamientos o metodologías que no necesariamente son obligatorios pues solo están indicándole al equipo de desarrollo cual debería ser el camino a seguir, dado que los equipos pueden tener un estudio más concreto que les pueda señalar un camino completamente diferente y de mejores resultados.

Esta disciplina tiene un componente de importancia muy alto a la hora de trabajar en el desarrollo de productos software, dado que como lo indica cada una de sus definiciones ayuda a los grupos de desarrollo a encaminarse de la mejor manera en el proceso de desarrollo y de esta manera al final de dicho proceso se logre obtener un producto que cumpla con las especificaciones deseadas y que sea de satisfacción para el cliente/usuario final.

Además porque aunque no es estrictamente necesario ceñirse a lo que indiquen las metodologías de desarrollo que están ligadas a la Ingeniería de Software, estas sirven de soporte para llevar de la mejor manera cada uno de los proyectos para conducirlos al éxito, pero en muchos de los casos se entenderá que no deba ser necesario tenerlas en cuenta, y es allí donde se puede correr el riesgo de que el proyecto llegue a fracasar, y es algo que podría ocurrir pero en definitiva no es para lo que se trabaja.

De manera genérica para el proceso de desarrollo de software se identifico que se poseen una serie de tareas las cuales están distribuidas dentro de etapas, que permitirán de cierta manera despejar el camino al grupo de desarrollo, con el fin de mejorar la productividad de este y también para que el producto final tenga la calidad necesaria.

1.2 PROCESO DE DESARROLLO DE SOFTWARE

En el marco de una metodología "*genérica*" si se le puede dar ese nombre, se identifico que existe una seria de etapas que marcan la pauta de cómo se debería llevar a cabo todo el proceso para llegar a obtener los resultados deseados, dichas etapas se enumeran a continuación:

1.2.1 Análisis de Requisitos. Es la etapa que tiene como una de sus tareas realizar la elicitación de cada uno de los requisitos que debe cumplir el sistema a desarrollar y además es la fase inicial del proceso, dado que esta es en gran medida quien brindara el horizonte en el que se debe encaminar todo el proceso de desarrollo del producto. Adicionalmente la fase del análisis de requisitos se es de suprema importancia porque es donde se debe obtener una versión final de lo que espera el cliente para satisfacer su necesidad, donde se discrepen elementos que no sean de mayor relevancia o que puedan complicar el buen desarrollo del proyecto. Por todo lo anterior descrito la etapa de análisis de requisitos tiene su objetivo principal enmarcado por la detección de conflictos y defectos en cuanto a la comunicación que se da entre el usuario y el personal del equipo de desarrollo, de tal manera que estos permitan ser transformados apropiadamente en elementos a tratar dentro del diseño.

1.2.2 Especificación. Esta etapa define de acuerdo al análisis previo de requisitos, el comportamiento que se espera del producto una vez terminado, de tal manera que supla las necesidades del negocio teniendo en cuenta las interacciones del desarrollo con los usuarios del sistema para la clasificación y especificación de dichos requisitos.

Para la realización de la especificación se podrían utilizar técnicas como los casos de uso que son básicamente la descripción grafica de los actores y actividades a realizar en el proceso, otra técnica que se utiliza es la de historias de usuarios, de las cuales hablaremos más adelante, son un poco mas informales y comúnmente utilizadas en las metodologías ágiles, caracterizada por la utilización del lenguaje común del usuario.

1.2.3 Arquitectura. Es la etapa donde el arquitecto como uno de los roles definidos dentro del proyecto, se encarga de estructurar cual será el diseño de la arquitectura a tener en cuenta dentro del desarrollo (Etapa de Programación), todo esto basado en elementos fundamentales como lo son los procesos de negocio, en el cual se podrá visualizar la interacción entre las entidades de negocio y por ende validar ese esquema arquitectónico, el cual está construido en esta etapa y debe estar acorde con lo que se definió en el alcance del proyecto, la complejidad del mismo y las necesidades establecidas.

En esta etapa el arquitecto tiene como soporte todo lo que es conocido como diagramas utilizando cualquiera de las herramientas o maneras de soportarlos, donde la herramienta más común y utilizada es la conocida como UML (*Unified Modeling Language*) y será esta persona la

que tenga a cargo la tarea o el rol de definir, de acuerdo a su experiencia y conocimiento, cuales son los diagramas más apropiados para dar a entender y conocer el esquema arquitectónico de la solución.

1.2.4 Programación. En esta etapa se requieren ciertas destrezas entre los participantes y al igual que cada una de las otras, es crucial dentro del proceso de desarrollo de software, dado que en esta es en la que se realiza la compleja traducción de todo lo previamente descrito en las otras etapas, a través de un lenguaje de programación (Java, C++, .net, entre otros), se dice que en esta etapa no se debería demandar mayor tiempo del proyecto y tampoco es vista como la más compleja, esto debido a que se piensa que dicha complejidad está dada al lenguaje de programación utilizado y a la correcta estructuración de cada uno de las etapas anteriores a esta, pues un deficiente análisis de requisitos y diseño de la solución conllevaría a una deficiente programación que culminara con la entrega de un producto defectuoso o en algunos casos con una concepción completamente diferente a lo que el cliente deseaba.

1.2.5 Prueba. Esta es la etapa que como su nombre lo indica es en donde se deben realizar las pruebas necesarias que permitan obtener los resultados en el mejor de los casos positivos del buen funcionamiento del producto final y de esta forma observar el cumplimiento de cada una de las especificaciones dadas en la etapa de análisis de requisitos. Dichas pruebas tienen diferentes técnicas y al mismo tiempo pueden realizarse de diferentes maneras; una de ellas es la de realizar pruebas independientes a cada uno de los módulos del producto para ver así su desempeño y posteriormente realizar una prueba conjunta de los mismos, y de esta forma poder llegar al objetivo trazado. En la etapa de pruebas se considera realmente importante que dichas pruebas sean realizadas por personas completamente diferentes a los involucrados en el desarrollo (Etapa de Programación) no dejando a un lado que estos últimos deben realizar las pruebas pertinentes antes de entregar el producto a la etapa de pruebas lo cual permitirá disminuir la cantidad de bugs a encontrar.

En esta etapa se habla que las personas que deben conformar el grupo de pruebas debe estar conformado con por lo menos una persona que tenga desconocimiento en el ámbito de testing dado que a través de ella se podrá tener una percepción (evaluación) de cómo es la percepción en cuanto a la documentación y que los procesos sean claros para el usuario a la hora de realizar cada una de las pruebas necesarias, además de personas que tengan el conocimiento suficiente en este ámbito para de alguna manera ir con mayor claridad a encontrar los posibles fallos que pueda tener dicho producto software.

1.2.6 Documentación. Esta etapa hace referencia a la recolección de toda la información que se ha generado en cada una de las etapas del proyecto, también hacen parte de esta los manuales tanto técnicos como de usuarios, todo esto con el propósito de un mayor entendimiento del sistema, de la usabilidad y poder a futuro mantener y ampliar el sistema si es requerido.

1.2.7 Mantenimiento. La fase de mantenimiento como su nombre lo indica es principalmente mantener el correcto funcionamiento del software y a su vez corrigen los errores (bugs), y se pueden incorporar nuevos requisitos para la adición de mejoras.
Gran parte del ciclo de vida del producto está destinada para esta etapa de tal manera que se puedan adicionar funcionalidades de acuerdo a los cambios que puedan surgir en el negocio o su mismo entorno.

Cada una de estas etapas del proceso de desarrollo software son plasmadas en la **Figura 1**, y se muestra un esquema en cascada, en el cual está inmersa la metodología RUP y otras que en todo el medio del desarrollo de software se aplican.

Figura 1. Fases del Proceso de Desarrollo de Software

kayingsoft.blogspot.com

La imagen anterior representa el proceso de desarrollo de software, pero en ciertas ocasiones esta ilustración puede sufrir modificaciones debido a que en algunos proyectos puede diferenciarse el análisis y la arquitectura.

1.3 METODOLOGÍA RUP (RATIONAL UNIFIED PROCESS)

Este es un proceso de la Ingeniería de Software que brinda una vista en detalle como en conjunto con la disciplina realiza la asignación de tareas y responsabilidades dentro de una organización que no necesariamente es de desarrollo.

Dicho proceso tiene como objetivo principal de brindar un aseguramiento en la construcción de software de alta calidad que de alguna u otra manera cumpla con cada una de las necesidades del cliente, teniendo presente dos aspectos importantes como lo son los costos y el calendario, siendo estos predecibles.

1.3.1 CARACTERISTICAS DE RUP. Dentro de la metodología RUP se lograron identificar una serie de características que pueden dar claridad a lo que esta se refiere. Dichas características se describen a continuación:

- Es una metodología guiada, todo esto dado que se esta soportada sobre diagramas uml que son de gran apoyo para los usuarios finales, pues son una manera de dar claridad respecto al comportamiento del software

- Es iterativo e incremental, esto debido a que divide su proceso en cuatro fases (Iniciación, Elaboración, Construcción y Transición), con el fin de realizar una serie de iteraciones en cada una de estas, de acuerdo al tamaño del proyecto y enfocarse en actividades que puedan ser determinantes a la hora de que resulte exitoso.

Donde este esquema se va trabajando el desarrollo basado en componentes, para así estar bien articulados en lo que corresponde a divide y vencerás, dichos componentes serán entrelazados para generar el sistema, es por esto que se habla que esta característica permite que el sistema vaya construyéndose poco a poco mientras se desarrollan los demás componentes. Por otro lado la utilización de un lenguaje estándar, el cual sea de fácil entendimiento para cualquier persona que tenga acceso a dicha información.

1.3.2 ELEMENTOS METODOLOGIA RUP. Cuando se habla de Metodología RUP, se deben tener presente elementos o conceptos importantes que servirán de guía para desarrollar de la mejor manera cada una de las fases que posee esta misma, por tanto los elementos o conceptos que están ligados a esta metodología son:

- Roles: que son quienes definen el comportamiento y las responsabilidades que debe asumir una persona o un conjunto de personas, donde una persona puede estar asumiendo una diversidad de roles, a grandes rasgos son quienes dan respuesta a la pregunta ¿Quién?

- Actividades: estas se refieren básicamente a la acción de realizar algo especifico de acuerdo a un rol asignado y serán las personas que tengan dicho rol las encargadas de realizar dichas actividades, por tanto son estas las cuales nos den una respuesta a la pregunta ¿Como?, es decir el como se va a realizar o llevar a cabo la solución al problema planteado.

- Productos: cuando se habla de un producto se esta haciendo referencia a elementos que se consideran tangibles o que dan la certeza de un resultado del proceso que se llevo a cabo teniendo presente cada uno de los elementos que acá en este documento se plasman, elementos que estarán enfocados a irse construyendo para al final obtener el producto final, y son quienes nos darán respuesta a cuando el grupo de desarrollo se pregunta ¿Qué?, el que se va a realizar y que debemos ir construyendo para llegar a ese producto final necesario.

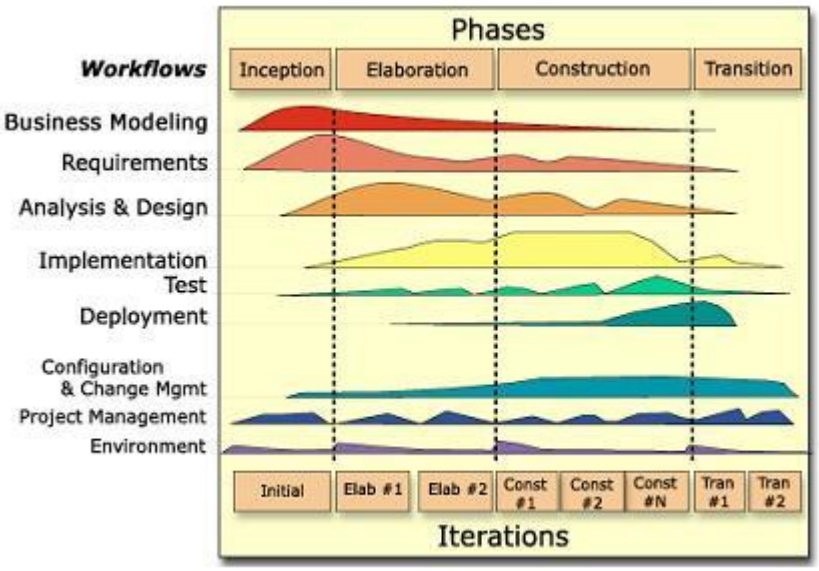
- Flujos de Trabajo: cuando nos hablan de flujos de trabajo o workflows, básicamente nos están indicando que con realizar una definición o establecer unos roles, unas actividades y unos productos estamos listos para enfocarnos en el desarrollo, pero esto es aun un poco más grueso de lo normal dado que para llegar a unos buenos resultados se debe tener en cuenta que existe una relación entre cada uno de los roles dado que cada uno de ellos estará llevando a cabo una serie de actividades que puede afectar directa o indirectamente a otro rol y es por esto que se debe establecer un esquema de navegabilidad donde esté dando a conocer cuál será la el camino a seguir representado en un lujo, y serán estos los cuales darán respuesta a la pregunta ¿Cuando?, es decir el cuando entra cada uno de los roles a ejercer sus funciones y generar los productos o artefactos necesarios.

Como se menciono anteriormente, dicha metodología en estos momentos posee un esquema dividido en cuatro fases que nos brindaran una visualización y un desarrollo del proyecto más eficiente y un mejor manejo para cumplir con los objetivos trazados.

1.3.3 FASES DE RUP. Al igual que las demás metodologías RUP posee una serie de fases que son las que permitirán tener un direccionamiento o alineamiento con el desarrollo del producto software, teniendo siempre en pie la entrega de algo que resolverá una necesidad a un usuario y que de alguna u otra manera estará brindando un valor agregado al mismo, por tal motivo se presentan cada una de las fases que hacen parte de RUP.

La Figura 2 nos muestra una visión grosso modo de cada una de las fases y workflows que posee la metodología RUP.

Figura 2. Metodología RUP



malkamasco.blogspot.com

Como se describió y se ilustro anteriormente con la figura cada una de las fases son:

- Iniciación (Inception), es quizás el pilar para un buen desarrollo de un proyecto, pues es en esta fase donde se realizan estudios necesarios para darle una visión desde diferentes puntos de vista y así determinar la viabilidad del mismo para la comunidad, pues es en esta fase donde vamos a entender cual es el enfoque o el objetivo al que le apunta la construcción del producto software, es decir en dicha fase se debe realizar un análisis bien detallado y lo suficientemente necesario para tomar una decisión al respecto del mismo.

  Los principales objetivos que se deben tener en esta fase están ligados a conocer el ámbito del conocer el entorno en el que se va a mover el proyecto, adicional a esto delimitar el área de este, establecer un marco financiero del proyecto muy a grandes rasgos, generar un posible esquema arquitectónico y realizar una evaluación muy general en el tema de riesgos. Es en esta fase donde se intenta definir todo el modelo de requisitos, pero es también allí donde se debe dejar plasmado la aprobación por parte de todas las personas interesadas en cada uno de los aspectos mencionados ya que se debe contemplar la posibilidad de realizar una redefinición, desechar la posibilidad de llevar a cabo el proyecto o de continuar en el caso de que así lo amerite.

- Elaboración (Elaboration), es la fase en la cual se va a analizar el problema con una profundidad aun mayor, establecer cual será su arquitectura y por ende construir un plan de trabajo que deberá llevarse acabo para brindarle satisfacción al cliente, pero sin dejar de lado la constitución de la mitigación de los riesgos en caso de que alguno de ellos se

pueda llegar a materializar. Además en esta fase se deben realizar actualizaciones a cada uno de los artefactos desarrollados en la fase anterior.

Al finalizar esta fase al igual que en la anterior se debe llevar a cabo un estudio de los elementos o artefactos para determinar la continuación o una posible redefinición de aspectos que son de vital importancia para un buen desarrollo del proyecto.

- Construcción (Construction), en dicha fase su principal objetivo esta dado a generar una operatividad para la construcción del producto que sea de manera incremental generado a través de una serie de iteraciones, es por esto que todos los elementos a desarrollar deberán ser llevados a cabo teniendo en cuenta cada una de las características y haber pasado por un proceso de pruebas lo cual nos permita llegar a un producto parcial que pueda ser entregado al cliente como una versión inicial para que el cliente vaya realizando pruebas mas exhaustivas o vaya brindando su punto de vista al respecto, claro que no solo en este aspecto sino que es acá donde se debe tratar de identificar la manera de optimizar cada uno de los recursos para tener una utilización adecuada y que nos permita disminuir costos pero teniendo siempre una calidad sostenida o en su defecto elevándose cada día mas.

- Transición (Transition), que es la fase que básicamente se encarga de realizar ese traspaso del producto a las manos del cliente o usuario final , donde sus principales actividades están ligadas a establecer lo que posiblemente llegaran a ser desarrollos de nuevas versiones donde se visualice una evolución del producto y que brinde un valor agregado a los usuario, adicional a esto se establece allí la completacion de la documentación, el ajuste, la configuración, la instalación y usabilidad del producto, todo esto a través de una transferencia de conocimiento a los usuarios para que estos tengan un mejor acercamiento con el producto final.

  En esta fase se busca que el usuario ya pueda de alguna u otra manera interactuar de manera sencilla con el producto y que este ultimo cumpla con cada una de las especificaciones dadas por el usuario a través de los requisitos expuestos en la fase inicial.

En la metodología RUP se tienen prestablecidos una serie de flujos de trabajo que son distintos entre si, pero que a su vez son desglosados en dos sub grupos denotados como s*ubgrupo de ingeniería*, dentro de este se pueden establecer flujos de trabajo como:

- Modelado del Negocio
- Requisitos
- Análisis y Diseño
- Implementación
- Test
- Despliegue

Adicional se encuentra el *subgrupo de apoyo*, en el cual se logran identificar flujos de trabajo tales como:

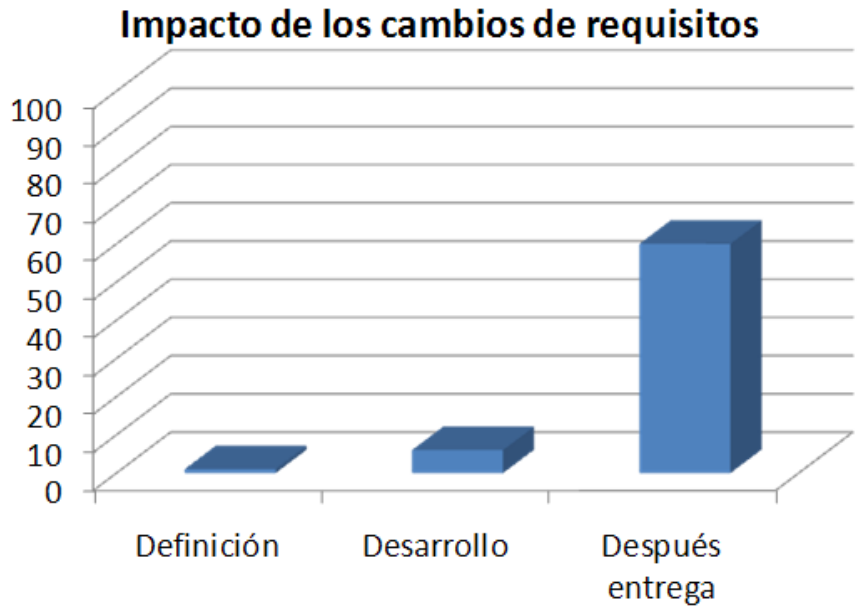
- Administración del Proyecto
- Configuración y Control de Cambios
- Entorno

Claro que muchos de los analíticos de todo el entorno de las metodologías llegan a pensar que cada uno de estos flujos está reconociendo a algunas etapas de las metodologías en cascada, sin embargo en esta metodología (RUP) se trabaja en un esquema completamente diferente, dado que se involucran cada uno de los flujos transversalmente con el proceso.

La metodología RUP posee una serie de ventajas sobre las demás metodologías, ya sean ágiles o tradicionales, todo esto dado que esta metodología se encarga de basarse en las mejores practicas que se encuentren relacionadas con el entorno, además de generar una iteración por cada una de las fases que tiene establecidas dando como resultado al finalizar las iteraciones un producto o insumo que será utilizado en la siguiente fase.

Dentro de todo el marco del proceso de la Ingeniería de Software cada una de las fases es vital para entregar al final un producto lleno de calidad y cumpliendo con cada una de las especificaciones dadas por el usuario a través de los requisitos, pero son estos últimos quienes desde el inicio te van a indicar cuál podría ser el camino más indicado para llevar a cabo el proyecto, es por esto que el cambio de estos en cualquiera de las fases del proceso estará generando un impacto mayor, mediano o pequeño según sea el cambio y el instante de tiempo en el que se encuentre el proyecto, la figura 3 que esta a continuación nos da un pequeño marco de referencia de cual podría llegar a ser el impacto al cambiar los requisitos.

Figura 3.Impacto de los Cambios de Requisitos



opinadeti.wordpress.com/

Es por esto que se le debe dar un manejo adecuado a toda la fase de elicitacion de requisitos, ya que esta sentara las bases para una muy buena construcción del producto software a entregarle a todos los usuarios y satisfacer una necesidad que ellos tienen en su momento.

2. INGENIERÍA DE REQUISITOS

Cuando se habla de la Ingeniería de Requisitos, se debe en principio tener claridad con lo que se puede entender o interpretar acerca de ¿Qué es un requisito?, por ello se encontraron y se exponen varias definiciones que se consideran apropiadas para definir este concepto.

“Un requisito es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”.

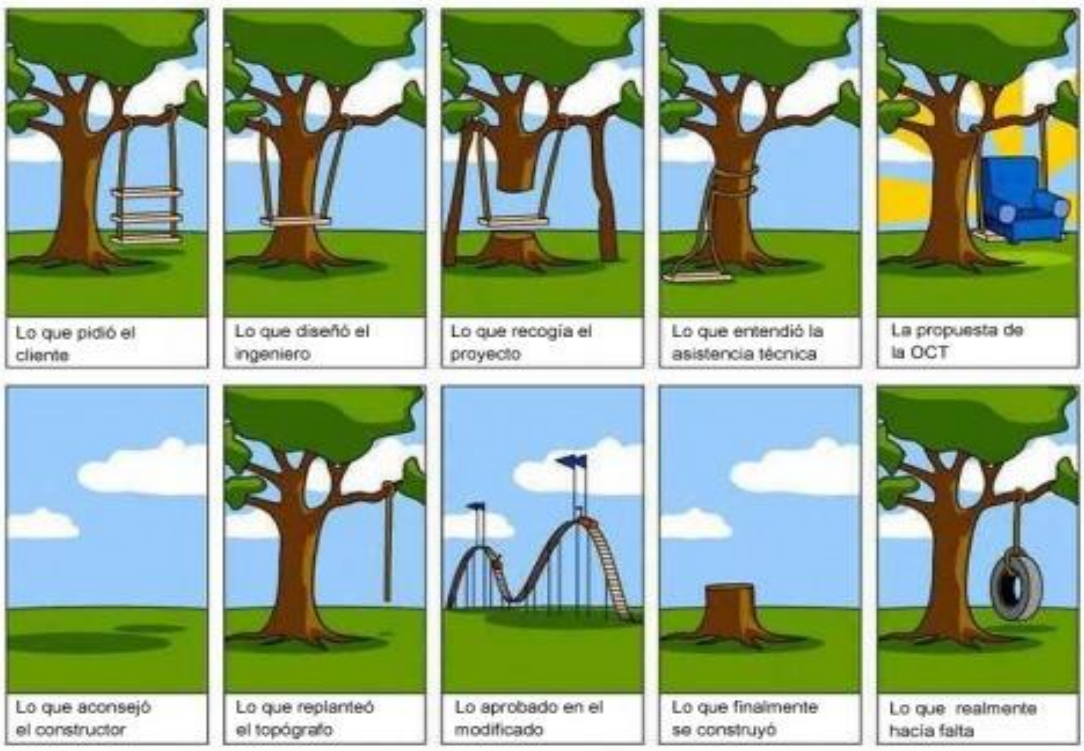
“Un requisito es una condición o capacidad que debe estar presente en un sistema o componente de sistemas para satisfacer un contrato, estándar, especificación u otro documento formal”.

De acuerdo a las definiciones anteriores los requisitos se entienden como una especificación de lo que se denota o se establece como necesario e indispensable para determinar el comportamiento eficiente de un sistema, tanto a nivel global como también de forma precisa en cada uno de sus atributos.

Estos mal manejados pueden convertirse en un obstáculo para el proceso de desarrollo del sistema, dado que son los usuarios los que están brindando cada una de sus necesidades en sus propias palabras, es acá donde se debe tener en cuenta cada una de las reglas de negocio y los objetivos tanto de la compañía como del software a construir, pues serán los requisitos los que tracen el camino de manera determinante a el proceso de desarrollo del mismo.

Es por esto que en muchos de los proyectos una mala interpretación en la elicitacion de los requisitos nos puede llevar a desarrollar un producto muy alejado de la realidad y que al finalizar el desarrollo va a tener un contraste demasiado grande e irreversible de acuerdo a lo que el usuario tenía en mente, claro está que en algunas ocasiones puede no ser una mala interpretación del integrante del equipo sino de las personas que están suministrando la información, es por esto que la Figura 4 ayuda a ilustrar y entender que pasa en la vida real referente a este aspecto.

Figura 4. Ingeniería Software



eljuguito.wordpress.com

2.1 CARACTERISTICAS DE LOS REQUISITOS

En dicha Ingeniería, el objetivo principal es que cada uno de los requisitos que se obtengan sean los adecuados y necesarios para darle una visión clara de lo que es el producto final, pero para que un requisito sea por si solo adecuado o tenga la mejor descripción de lo que quiere el cliente, debe satisfacer una serie de características las cuales se describen a continuación:

2.1.1 Necesario. Esta característica quiere decir que represente en realidad lo que es necesario para el producto, es decir que lo que se exprese en dicho requisito es necesario para el buen desempeño y desarrollo del producto.

2.1.2 No ambiguo. Es decir debe diferenciarse de los demás requisitos además de tener claridad en su interpretación independiente del punto de vista que se visualice.

2.1.3 Conciso. Donde básicamente debe expresar lo que se desea de una manera breve y comprensible, que permita facilitarles su entendimiento al equipo desarrollador y al cliente para obtener un excelente producto.

2.1.4 Completo. Como su mismo nombre lo dice debe contener la información requerida para un buen entendimiento del mismo.

2.1.5 Alcanzable. Debe ser algo muy puesto sobre la tierra, el cual tenga un objetivo alcanzable a través de diferentes recursos.

2.1.6 Verificable. Que pueda brindarle a los usuarios la certeza de que fue cumplido el requisito a través de un componente del desarrollo software o de una funcionalidad.

2.2 TIPOS DE REQUISITOS

La Ingeniería de Requisitos, como su mismo nombre lo indica tiene como elemento fundamental el requisito que como está plasmado anteriormente es la expresión de una necesidad que tiene el cliente, sin embargo este puede ser distribuido en diferentes tipos de requisitos, unos más utilizados que otros, pero dicha clasificación se especifica a continuación:

2.2.1 Requisitos Funcionales. Son los que de alguna manera u otra describen las funcionalidades o el comportamiento que debe tener el sistema, estos requisitos vienen a ser complementados por cada uno de los requisitos no funcionales, en conclusión estos son los que constituyen el comportamiento del sistema y es por tal motivo que la definición de cada uno de ellos debe provenir de la interacción con los clientes/usuarios, proveedores o personas involucradas en el desarrollo del software y además se deben tener en cuenta cada una de las reglas de negocio que fueron establecidas en la organización, ya que estas brindan directrices con respecto a cómo está establecido el negocio.

2.2.2 Requisitos No Funcionales. Son aquellos que determinan una necesidad en el sistema pero que no hacen parte de una funcionalidad específica que se tenga que construir en el sistema, es por esto que hacen referencia contrariamente a elementos de información o de funciones que deba cumplir el sistema.

2.2.3 Requisitos de Información. Son aquellos que a diferencia de los anteriores va más de la mano de lo que se denominan restricciones, son elementos de información que le brindan al equipo de desarrollo unas restricciones a tener en cuenta a la hora de trabajar en el producto, es decir elementos que directa o indirectamente afectan la labor de desempeño.

Luego de tener claridad en lo que es un requisito, se puede dar un vistazo a lo que es la Ingeniería de Requisitos, donde esta es la que agrupa cada una de las tareas que van ligadas a lo que son las

necesidades o condiciones para satisfacer dentro de un desarrollo software, donde dichas necesidades o condiciones son denominadas requisitos como se expreso anteriormente.

Y es esta ingeniería la que se encargue de obtener una buena especificación de los requisitos, dado que estos sean medibles, comprobables, sin ambigüedades entre otros aspectos, para que de esta manera la fase a seguir en el proceso de desarrollo de software vea con mayor claridad el camino a seguir recorriendo para obtener al final un buen producto software que le brinde satisfacción al cliente y usuario final.

La Ingeniería de Requisitos posee varias implicaciones en cuanto a las actividades dentro de su ciclo de vida entre las cuales podemos ver que es necesario un análisis del problema, una fase de elicitacion de requisitos, para posteriormente realizar un análisis de los requisitos identificados y realizar los cambios pertinentes a los mismos, en caso de ser necesario o en su defecto descartar alguno, y luego de obtener la lista depurada o especificada de los requisitos a tener en cuenta, se generara una priorización de los mismos para determinar cuál es su peso o importancia dentro del proyecto, en caso de existir algún conflicto entre requisitos la negociación es lo más adecuado para llegar a un feliz término entre esos conceptos, y así mismo debe implicar una etapa de verificación y validación de cada uno de los requisitos para en el final analizar su rastreabilidad.

2.3 PROCESO DE LA INGENIERÍA DE REQUISITOS

A lo largo del proceso de la Ingeniería de Requisitos se encuentran una serie de procedimientos, que se convierten en los lineamientos del correcto uso de esta ingeniería, su aplicación rigurosa es un factor determinante para la elaboración de productos con alta calidad, a continuación se exponen unas breves definiciones de las etapas de dicho proceso:

2.3.1 Análisis del Problema. Cuando se habla de Requisitos se habla de antemano de una situación o de un problema en particular, que requiere un previo análisis de todo el entorno e implicaciones que se evidencien para reconocer cada uno de los elementos encontrados, de la misma manera como el usuario los percibe y los entiende dando mayor claridad y acercamiento a cual es realmente el problema que se quiere resolver y obtener toda la información oportuna, lo cual permitirá una mayor preparación para las fases siguientes.

Si se tiene un buen conocimiento del negocio se pueden plasmar más fácil los requisitos. Es por esto que dentro de esta etapa es ideal conformar el grupo de personas del negocio que estarán involucradas de inicio a fin con el proyecto de desarrollo y así de esta manera lograr el cumplimiento de cada uno de los objetivos trazados. Es de vital importancia identificar a las

personas con las mejores capacidades, pues serán ellas quienes se encarguen de mostrar el horizonte del proyecto.

2.3.2 Elicitación de Requisitos. Para esta etapa del proceso de Ingeniería de requisitos se requiere una buena preparación y habilidades sociales y psicológicas para comprender a los actores implicados en el proyecto y considerar las necesidades de cada uno y de esta manera generar las bases solidas de lo que se necesita Construir.

Existen diversas técnicas de elicitación entre las más comunes se encuentran las entrevistas, Talleres, Brainstorming, workshops:

- Entrevistas: esta es tal vez la más común de todas y por lo general se entrevista solo en pequeños grupos de trabajo, pueden ser estructuradas en las cuales se define una agenda de preguntas abiertas para los participantes o abiertas las cuales no tienen una agenda común. Las ventajas de estas son que se puede obtener buena colección de información y descubrir deseos, metas, opiniones entre otras, claves para los requisitos del Sistema. Las desventajas podrían ser la predisposición de los entrevistados y el análisis de datos cualitativos
- Talleres: es común que los requisitos tengan diferentes puntos de vista de acuerdo a las necesidades de cada implicado o usuario lo cual hace conveniente la utilización de los talleres para que de manera controlada se generen discusiones entre los implicados, que permitan obtener detalles y por ende requisitos insospechados.
- Lluvia de ideas (Brainstorming): esta técnica de elicitacion de requisitos se realiza de manera grupal y permite en un entorno relajado y natural para los implicados aportar sugerencias creativas libremente, incentivando el surgimiento de nuevos requisitos aprovechando la capacidad creativa de manera oportuna en un ambiente propicio. Sus etapas suelen ser la preparación, generación, consolidación y documentación.
- Forma de contrato: consiste en llenar formularios o contratos indicando los requisitos. Puede resultar extenso de acuerdo al tamaño del Proyecto y por ende tendera a ser tediosa su utilización dentro del desarrollo del software.
- Objetivos medibles: es una técnica que tiene como principal objetivo recopilar todos los requisitos que se convierten en objetivos generales pero que posteriormente se analizaran detalladamente con el fin de establecer objetivos concretos que estarán de la mano con lo que el usuario expresaba que deseaba que el sistema realizara, además se establecerán mecanismos a través de los cuales se dé a conocer el progreso de estos.
- Prototipos: con esta técnica lo que se pretende es darle al cliente una primera visión de lo que ellos quieren que haga el sistema, los prototipos son un ejemplo que muestra a grandes rasgos como se podría visualizar el producto desarrollado y de esta forma ver si se está enfocado en lo que es o si se está desviado de la realidad.

- Casos de Uso: esta por el contrario y al igual que la ultima, es utilizada para representar los que son catalogados requisitos del sistema, pues es a través de esta técnica que se documentan cada uno de estos requisitos elicitados y que serán pieza clave a la hora de realizar la implementación del software.

2.3.3 Análisis de Requisitos. Esta es una etapa en la cual se debe tener presente cada uno de los requisitos elicitados en la etapa anterior, es necesario ponerlos en discusión para que de esta manera se puedan encontrar posibles falencias en las definiciones de dichos requisitos para proceder a sus correcciones y así entender lo que el usuario necesita y poderlo expresar en palabras más técnicas que sirvan de la mejor manera a la hora de establecer un diseño y su posterior construcción.

2.3.4 Priorización de Requisitos. En el proceso de la ingeniería de requisitos, cuando ya se tienen establecidos los requisitos y previamente analizados, es necesario trabajar muy de la mano con el cliente para de alguna u otra manera se puedan establecer una serie de prioridades sobre esta lista de requisitos, que le permitan al equipo de desarrollo enfocarse en trabajar o enfocarse primero en aspectos de mayor urgencia, esto según lo considere el cliente y según el equipo de desarrollo lo vea viable, ya que en muchos aspectos por mas que se quiera trabajar primero en un enfoque este dependa de otros elementos que están relegados a trabajar posterior a estos.

En esta etapa al igual que en todas, las personas involucradas son parte fundamental para establecer un listado de requisitos que serán pieza clave en el rompecabezas a armar con el desarrollo software.

2.3.5 Negociación de Requisitos. Dentro de la ingeniería de requisitos, inclusive de las demás ingenierías siempre se van a encontrar posibles conflictos (en nuestro caso conflictos de requisitos), que deben ser tratados de la mejor manera y llegar a un acuerdo que permita una buena continuación del proceso de desarrollo que les permita llegar a un final esperado. Es en esta etapa donde se debe llevar a cabo las discusiones de los conflictos que fueron encontrados, analizar la problemática, establecer el proceso a seguir para culminar con una negociación entre las partes involucradas.

Todo esto con el fin de no entorpecer de alguna manera la continuación adecuada del proyecto.

2.3.6 Validación y Verificación de Requisitos. En el transcurso de esta etapa como su mismo nombre lo dice se realiza la validación y verificación de cada uno de los requisitos obtenidos en la elicitacion, paso fundamental para continuar con el proceso de desarrollo del producto software.

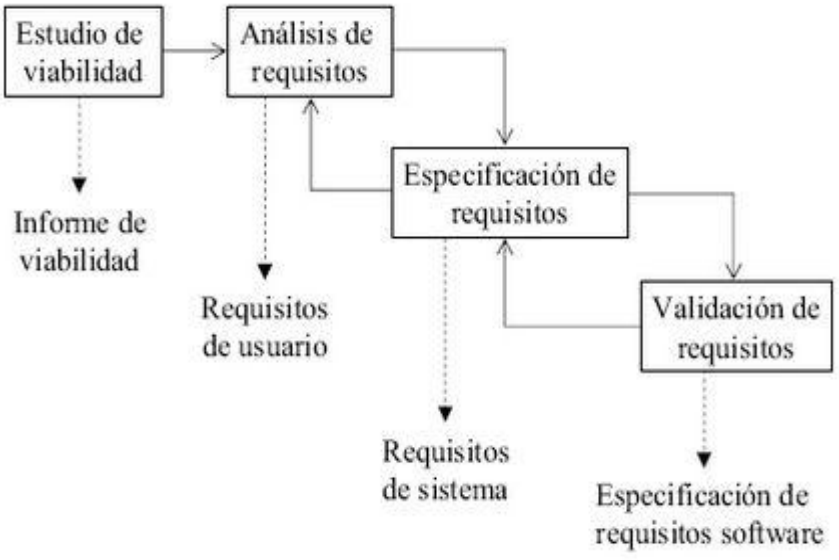
Cuando se habla de validación se está haciendo referencia a realizar un análisis adecuado para determinar si el trabajo que se está llevando a cabo esta cumpliendo con cada una de las necesidades que fueron hechas por el usuario.

Mientras que cuando se habla de verificación se hace referencia a que luego de realizar el análisis podamos preguntarnos si efectivamente se está construyendo el producto de la manera correcta para obtener buenos resultados.

2.3.7 Rastreabilidad. Esta etapa consiste en la forma de dejar huella o rastro de la realización de cada uno de los requisitos de tal forma que en determinado momento si es necesario se pueda devolver para encontrar posibles incorrecciones en la especificación de los requisitos elaborados.

En la **Figura 5** se puede identificar todo un esquema de todos los pasos que se deben seguir en las etapas de elicitación y análisis de requisitos, este simplemente indica las tareas de allí deben cumplirse.

Figura 5. Tareas etapas de Elicitación y Análisis de Requisitos



es.wikipedia.org/wiki/Software

3. METODOLOGIAS ÁGILES

Desde hace algún tiempo en el mundo del desarrollo de software han estado surgiendo algunas notaciones en el modelado que permiten una mejor visual de lo que implica este desarrollo, claro que no solo fueron las notaciones sino también que tras de estas fueron saliendo a relucir, dándose a conocer herramientas que pretendieron ser pieza fundamental para que el desarrollo de software tuviera éxito, pero desafortunadamente en dicho tiempo las expectativas no fueron satisfechas. Esto fue dado a raíz de que en esos momentos todo el concepto de Metodología Ágil había tenido una parada significativa y es por eso que se habla de que no sirve de nada el tener un esquema de notaciones y herramientas si no se tiene en el momento una directriz para tener la aplicación de estas, por tal motivo es que en los últimos días se viene presentando un alto interés en cuanto a las Metodologías de Desarrollo nos referimos.

En el año 2001 surge el termino ágil para el ámbito del desarrollo de software, esto dado que se pretendía dar un esbozo de lo que son los valores y principios que deberían permitir un desarrollo de software aun mas rápido y con una reproducción de cambios que puedan ir surgiendo en el camino del desarrollo de proyectos, esto con el fin de dar una alternativa a las metodologías tradicionales que son caracterizadas por ser metodologías rígidas y dirigidas por la documentación, que se genera en cada una de las actividades dentro de las fases del proceso del desarrollo de software.

Es en ese año que un grupo de personas, en su totalidad 17, que venían trabajando en todo el entorno de desarrollo software con metodologías tradicionales, se dan a la tarea de darle el nacimiento al termino “ágil” enfatizado a este entorno y en el cual se estableció lo que en estos momentos se conoce como **The Agil Alliance**, una organización creada con el fin de promover a todas las personas que están dentro del medio del desarrollo de software, aspectos que se deben tener en cuenta cuando se requiere agilidad en el desarrollo, todo esto dio inicio a lo que hoy en día es conocido como *Manifiesto Agil*, tratado en el cual se refuerzan valores que deben ser fundamentales a la hora de realizar un desarrollo.

3.1 PRINCIPIOS

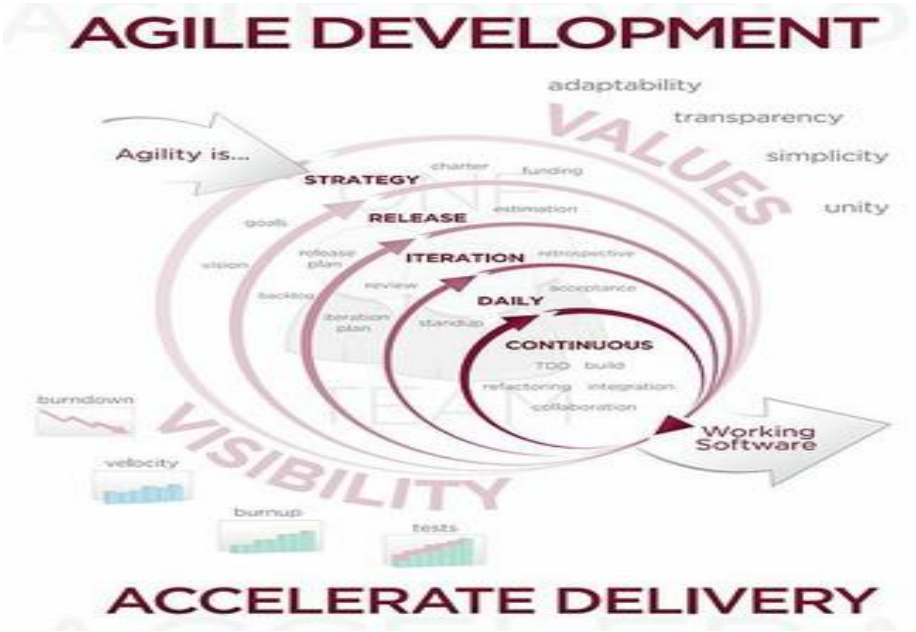
En el manifiesto ágil se reúnen los principios que sobresalen teniendo en cuenta cada una de las metodologías que existen y es por eso que surgen cuatro postulados que se describen a continuación:

- A los **individuos y su interacción**, por encima de los procesos y las herramientas.
- El **software que funciona**, por encima de la documentación exhaustiva.
- La **colaboración con el cliente**, por encima de la negociación contractual.
- La **respuesta al cambio**, por encima del seguimiento de un plan.

Y aunque en las metodologías tradicionales se le da mayor valor a cada uno de los que se encuentran a la derecha, en las metodologías ágiles se le da una mayor preponderancia a los elementos que se encuentran resaltados en la izquierda, debido a que se considera más eficiente realizar un desarrollo de dichas maneras.

Cada uno de estos principios se ve reflejado en la **Figura 6**, la cual representa el esquema de una metodología ágil, pues esta muestra como todo el desarrollo ágil esta mucho más enfocado a lo que es el valor para el desarrollo de software donde el cliente es un punto fundamental y la incorporación de alguna u otra manera de los cambios.

Figura 6. Desarrollo Ágil



www.ittera.net/metodologia

El desarrollo ágil de software se encuentra en el marco de lo que se denomina Ingeniería de Software, y de esta manera promueve realizar las fases por medio de iteraciones en cuanto al desarrollo a lo largo de todo el ciclo de vida del proyecto. Hoy en día existen muchos métodos de desarrollo ágil, los cuales en su mayoría minimiza riesgos desarrollando software en cortos periodos de tiempo. Cuando se habla que promueve iteraciones en el desarrollo esto nos quiere indicar que cuando el software es desarrollado en una unidad de tiempo este es llamado una iteración, la cual en algunos de los casos puede durar varias semanas, todo esto dependiendo de qué tan grande es el desarrollo y que tantas implicaciones puedan existir en cada paso de dicha iteración. Cada una de las iteraciones del ciclo de vida esta desagregado por varias fases, las cuales son: fase de planificación, fase de análisis de requisitos, fase de diseño, fase de codificación (desarrollo), fase de pruebas (revisión) y documentación. Cuando se da por terminada una iteración el equipo del proyecto realiza un análisis de las prioridades del mismo para de alguna manera u otra estructurar los pasos a seguir.

Se habla de que el termino de Desarrollo Ágil se remonta a los años 90's, dado que se veía a cada una de las metodologías tradicionales como métodos demasiado rígidos, estrictos y estructurados los cuales fueron heredados de la metodología en cascada. Es por eso que se dice que el proceso de desarrollo maneja un énfasis en cuanto al control del mismo a través de un esquema riguroso de definición de roles, actividades y artefactos, sin dejar de lado el modelado y una documentación detallada.

El esquema denominado "*tradicional*" a dado a conocer a todo el mundo que es de vital importancia su aplicación en proyectos de gran envergadura en cuanto a tiempo y recursos se refieren, donde se presenta una alta exigencia con respecto a la formalidad. Sin embargo muchos autores o personas expertas en el desarrollo hablan de que este enfoque no es muy aplicable sobre aquellos proyectos en los cuales el entorno del mismo es demasiado cambiante, ya que además exige trabajarlo en un periodo de tiempo mucho más corto de lo normal, teniendo en cuenta que se debe evidenciar una alta calidad en el producto final. Es por eso que en la gran mayoría de los casos los equipos de desarrollo al ver estos impedimentos para utilizar las metodologías tradicionales optan por hacer a un lado todo lo que predican estas últimas, con el fin de lograr acoplar las restricciones y asumiendo el riesgo que implica transitar por ese camino.

Es acá donde se habla que las Metodologías Ágiles surgen con el fin de dar respuesta a ese denominado gran vacío que se evidencia en las metodologías tradicionales, dado que están mucho más enfocadas a ser utilizadas en proyectos de un tamaño moderado en cuanto al volumen se refiere, es por eso que estas se visualizan como un solución a la medida para estos entornos, dando un vuelco total y generando una simplificación en el desarrollo del producto pero nunca dejando de lado la calidad que debe estar inmersa.

En los métodos tradicionales se presenta la documentación como un elemento fundamental a la hora de realizar la elicitacion de requisitos, pues se deben plasmar en el documento, el cual va a

servir de guía en todo el proceso de desarrollo del producto. Mientras que en las metodologías ágiles se tiene un propósito que es de vital importancia y es que el sistema o software que se entrega al cliente debe generarle valor a este, por tanto cada uno de los modelos y documentos generados durante todo el proceso de desarrollo se consideran elementos que no brindan un valor al cliente, por tal motivo se considera que si el fin de estas metodologías es optimizar el flujo de entrega de valor en cada uno de los desarrollos de software, se debería pensar muy seriamente en eliminar procesos que en estos momentos se consideran innecesarios.

Como en anteriores ocasiones se expreso, las metodologías ágiles han surgido con el fin de brindarle a los equipos de desarrollo una mayor agilidad en cuanto al transcurrir del proyecto, generando una menor inversión en dos aspectos demasiado importantes como lo son el tiempo y los recursos que impulsaran el paso a paso del mismo, claro está que teniendo siempre presente el esquema de calidad que para los usuarios es de vital importancia al igual que otros aspectos más. Esto debido a que se trabajaran los proyectos con una mayor velocidad respetando la calidad del desarrollo y reduciendo los costos que se puedan generar si de una u otra manera se utilizan las metodologías tradicionales pues estas implican una serie de actividades de mucho más trabajo y que requiere de mucho esfuerzo.

El objetivo principal de las metodologías de desarrollo ágil está centrado en desarrollar un sistema de manera incremental dentro de un entorno que sea colaborativo, esto es que todos los integrantes del equipo estén familiarizados con todo el entorno del proyecto, todo esto dado porque estas metodologías fueron construidas teniendo como base el principio de continuar aprendiendo sobre el sistema y todo el entorno que lo rodea durante la ejecución del proyecto.

Esto es lo que le permite al usuario estar mucho más involucrado con el desarrollo que se está realizando, pues es este quien estará probando el software después de terminada cada una de las iteraciones y va entregando cada una de las correcciones a esos bugs (errores), que serán corregidos en la siguiente iteración.

Dentro del marco de lo que son las metodologías ágiles y tradicionales se pueden presentar algunas diferencias entre unas y otras, por eso se considera que ambos esquemas se pueden diferenciar debido a que las metodologías ágiles tienden a ser muy adaptativas mas no predictivas, caso contrario ocurre con las tradicionales, dado que estas presentan una planificación desde el inicio del proyecto por un largo periodo de tiempo, donde se tiene un gran éxito siempre y cuando cada uno de esos elementos planificados desde el inicio no sufran cambios sustanciales, pues de esta manera dicha planeación debería tomar un rumbo completamente diferente. Con las ágiles se tiene una percepción diferente ya que estas fueron construidas para que se adaptaran a los cambios que fuesen surgiendo en el transcurso del desarrollo del producto software.

Por otro lado las metodologías ágiles están más enfocadas a las personas mas no en los procesos que se tienen establecidos en las organizaciones, sin apartarlos totalmente pues son pieza fundamental, es evidente que estas metodologías se aferran mucho más en las capacidades de las

personas, es decir en su experiencia, en sus competencias y en su colaboración para construir software de calidad, y además respecto a los procesos deberían brindar mayores beneficios a través de la documentación que permita a los nuevos participantes conocer más detalles.

En consecuencia cuando una organización o un grupo de desarrollo se mueve de utilizar metodologías tradicionales a usar metodologías ágiles, es allí donde se pierde el esquema por fases que tienen estas primeras, como lo son la fase de análisis, diseño, implementación y pruebas; todo esto debido a que en las metodologías ágiles dichas fases se ejecutan siempre en paralelo, lo cual implica una estructuración diferenciada del proyecto. Con respecto a los equipos de desarrollo, se considera más apropiado crear equipos que lleven a cabo cualquiera de las funciones establecidas, y no un equipo donde cada uno de sus integrantes tenga un rol definido o una fase en la cual se especialice.

3.2 PRINCIPALES METODOLOGIAS ÁGILES

Dentro de las metodologías ágiles podemos encontrar una gran variedad, dado que cada una tiene una manera particular de trabajar o de llevar a cabo el proceso de desarrollo de software teniendo en cuenta cada uno de los elementos discutidos dentro de estas metodologías, pero yendo por un camino completamente diferente para llegar al mismo destino brindándole una gran satisfacción al cliente. Como una gran diversidad de metodologías ágiles, se logro identificar las más reconocidas, entre las cuales podemos identificar:

SCRUM.

XP, Extreme Programming.

AUP, Agile Unified Process.

KANBAN.

Crystal Clear.

Lean Software Development.

Se identificó que existen diversidad de metodologías ágiles que hoy día son trabajadas y reconocidas, sin embargo se planteo dar una breve descripción de lo que son las dos metodologías ágiles más conocidas a nivel mundial y quizás las más utilizadas en el momento por cada uno de los equipos de desarrollo de software.

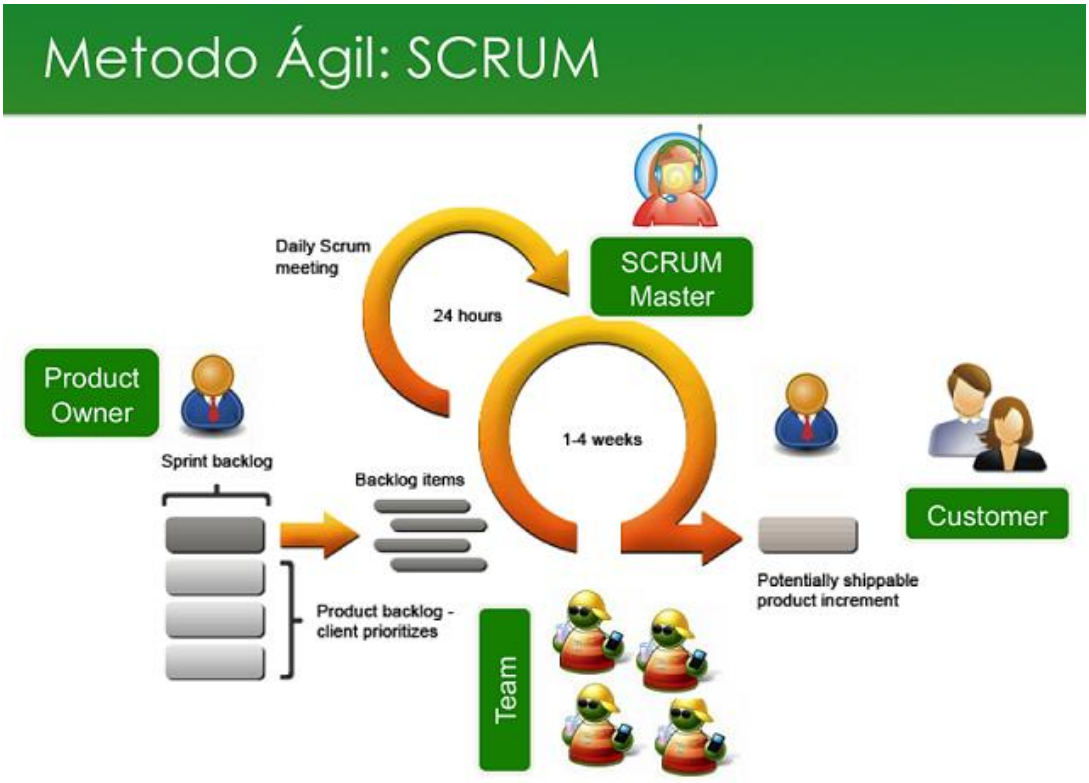
3.2.1 SCRUM. Metodologías como Scrum que se establece como un marco de desarrollo de software que esta soportado en un proceso de construcción de manera iterativa o incremental y se asemeja a todos los ambientes de desarrollo ágil. Dentro de esta metodología se pueden identificar tres roles como lo son ScrumMaster que es quien tiene una serie de funciones muy de la mano de las del director de proyecto, el ProductOwner que es quien representa a todos los stakeholders y el Team que está conformado por los desarrolladores, sin embargo existen otros roles que no son muy formales pero que no se dejan de lado como lo son los stakeholders donde podemos encontrar a los clientes, los proveedores, entre otros involucrados en el desarrollo de software, pero también los managers que en la gran mayoría son las personas que definen de alguna manera cual será el ambiente para el desarrollo del software.

Dentro de esta metodología se establece un concepto que es el sprint que se asemeja a lo que es para otras metodologías es una iteración, término que es establecido y estándar a la hora de trabajarla, donde el sprint puede estar durando entre una y cuatro semanas claro que esta magnitud es dada por el equipo de desarrollo (Team), dentro del sprint se tienen en cuenta varios elementos de gran importancia como lo son el Product Backlog, que concierne a lo que describe cada uno de los requisitos priorizados, el Sprint Planning que es la reunión en la cual de determinan cuales son los elementos que están en el Product Backlog y que quieren ver completados y el Team define cual va a ser la cantidad. El Product Owner es el grupo de personas que identifica dichos elementos del Product Backlog y que intervienen en el Sprint Planning, además de esto durante el transcurso del Sprint no se puede modificar el Sprint Backlog, lo que nos indica que los requisitos permanecen firmes en el transcurso del sprint.

En esta metodología ágil predomina el concepto de Product Backlog el cual está compuesto por las historias de usuario y posterior a esto se realiza una priorización que será puesta en común para el desarrollo del software. Claro que no solo se tienen elementos claves como las historias de usuarios o los Product Backlog, sino que también se encuentran las Daily Scrum, que vienen a significar las reuniones que se hacen cada día del sprint para identificar como es la evolución del proyecto, los Scrum de Scrum, los cuales son efectuados luego de la realización del Daily Scrum donde se discute del trabajo que se está realizando. Además de estos se encuentra lo que se denominan las Sprint Planning Meeting que es la reunión en la cual se realiza la planeación del próximo sprint y donde se estructurara el Sprint Backlog y demás detalles del proyecto.

La **figura 7** grafica de una mejor manera cada una de las explicaciones brindadas anteriormente, dado que se encarga de contextualizar al usuario del como trasciende el proceso de desarrollo con una metodología ágil como esta.

Figura 7. Metodología Ágil – SCRUM

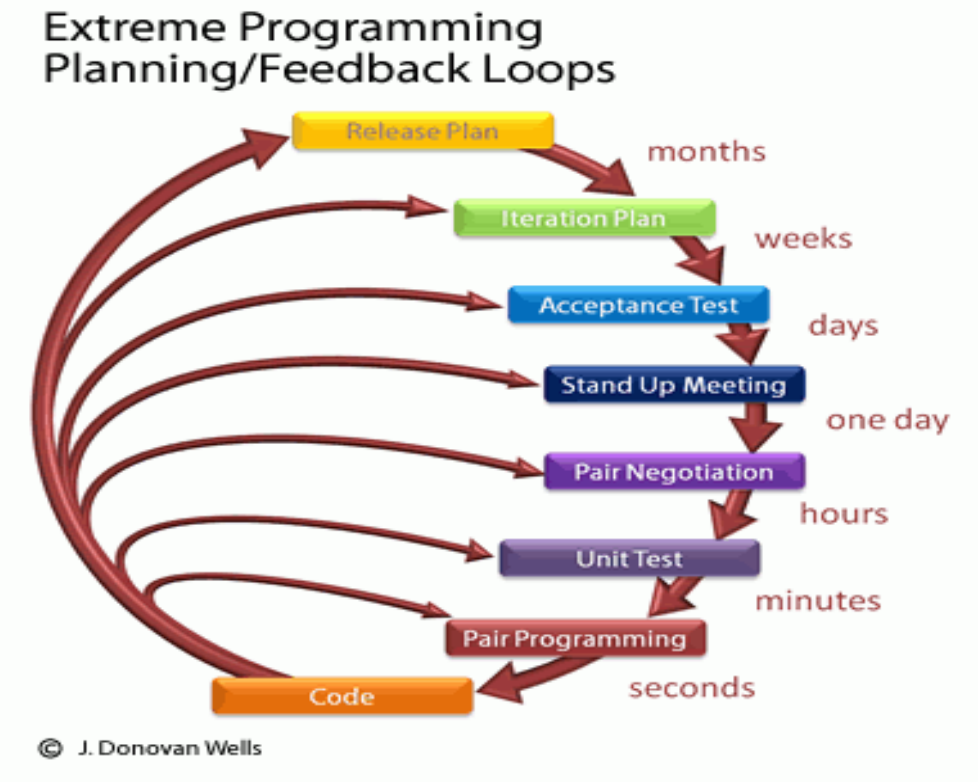


http://jsoftblog.wordpress.com/2012/02/09/metodos-ágiles-en-el-desarrollo-de-software/

3.2.2 XP, EXTREME PROGRAMMING. Otra de las metodologías es Extreme Programming (XP), que al igual que Scrum es una de las más reconocidas a nivel mundial y más que todo es dada a apuntarle a mejorar los desarrollos de software en el ámbito de la calidad de producto y de mejorar la capacidad de apoyar al cliente atendiendo las necesidades cambiantes del los mismos. Dentro de esta metodología las iteraciones son denominadas timeboxing y por el contrario de Scrum, en esta metodología es factible y permisible que el usuario brinde nuevos requisitos a ser adoptados.

Al igual que la metodología Scrum, tiene una serie de características que a grandes rasgos pueden verse similares pues ambas apuntan a tener presente valores como la simplicidad, comunicación, feedback, respeto entre otros, pues son demasiado importantes a la hora de tener un buen desarrollo y desempeño del proyecto, pero su principal característica es que se enfoca más en la adaptabilidad que en lo que se denomina previsibilidad.

Figura 8. Extreme Programming – Planning / Feedback Loops



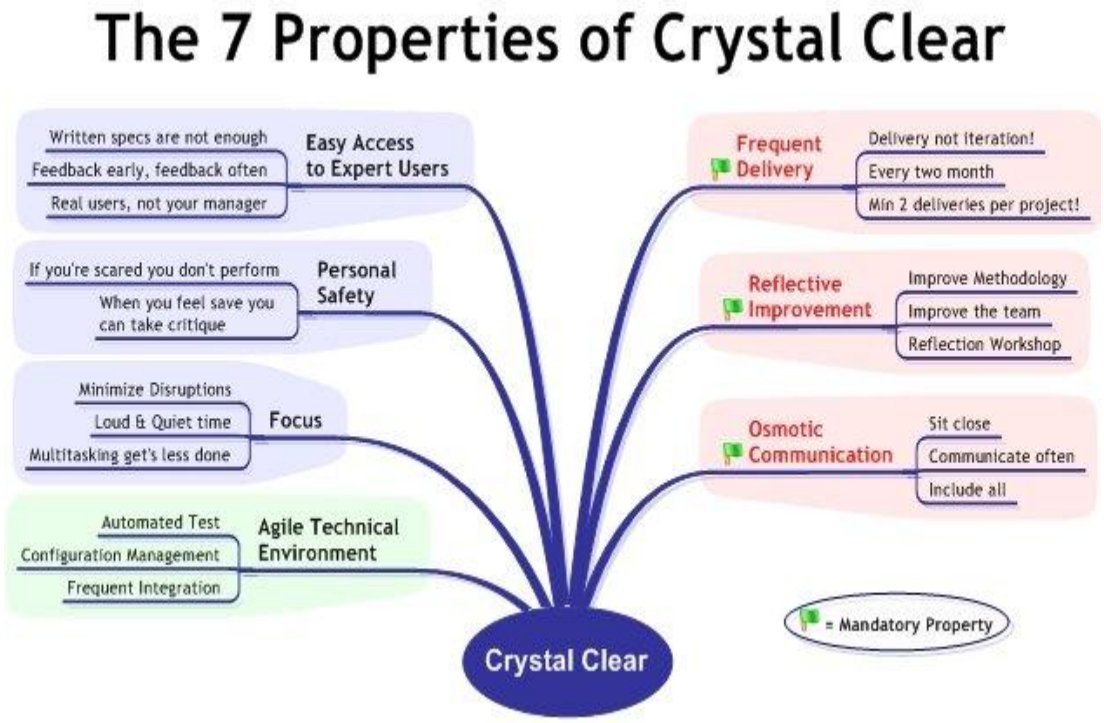
nosexybot.blogspot.com/2012/04/extreme-programming-xp.html

3.2.3 CRYSTAL CLEAR. Claro que no se puede dejar de lado otra de las principales metodologías ágiles, como lo es la Crystal Clear la cual al igual que las demás tienen un enfoque ágil y que además se enfoca demasiado en lo que es la comunicación pero que además es un poco más flexible con el tema de los entregables, pues en esta se manejan iteraciones que son demasiado cortas pero que presentan un feedback muy constante el cual permitirá ir generando correcciones tempranas en el desarrollo del software, es por eso que se implementa un rol de cliente/usuario que sea quien participe de manera activa con el tema de validaciones y verificaciones de cada uno de los requisitos funcionales, no funcionales y demás.

No solo las anteriores metodologías ágiles poseen un conjunto de valores con los cuales se soportan para realizar un buen trabajo, sino que Crystal también posee sus propios valores o propiedades como lo son, la entrega frecuente, comunicación osmótica, mejora reflexiva, seguridad personal, foco, fácil acceso a usuarios expertos y ambiente técnico con prueba

automatizada, managment de configuración e integración frecuente, donde cada uno de estos ayudara o apoyara de alguna manera para el desarrollo del proyecto.

Figura 9. The 7 Properties of Crystal Clear



http://www.wissel.net/blog/d6plinks/SHWL-6DVDFT

3.2.4 KANBAN. Esta es una metodología la cual tiene una historia muy peculiar, dado que estas surgen a través de un sistema de producción en una de las compañías mas grandes del mundo como lo es Toyota, dado que tienen presente cada uno de los principios que se plasmaron en el manifiesto ágil, dicha denominación como *Kanban* fue basado en las tarjetas que en aquella época fueron construidas para indicar una serie de instrucciones que permitiesen ir de lleno con el desarrollo Justo a Tiempo (Just in Time).

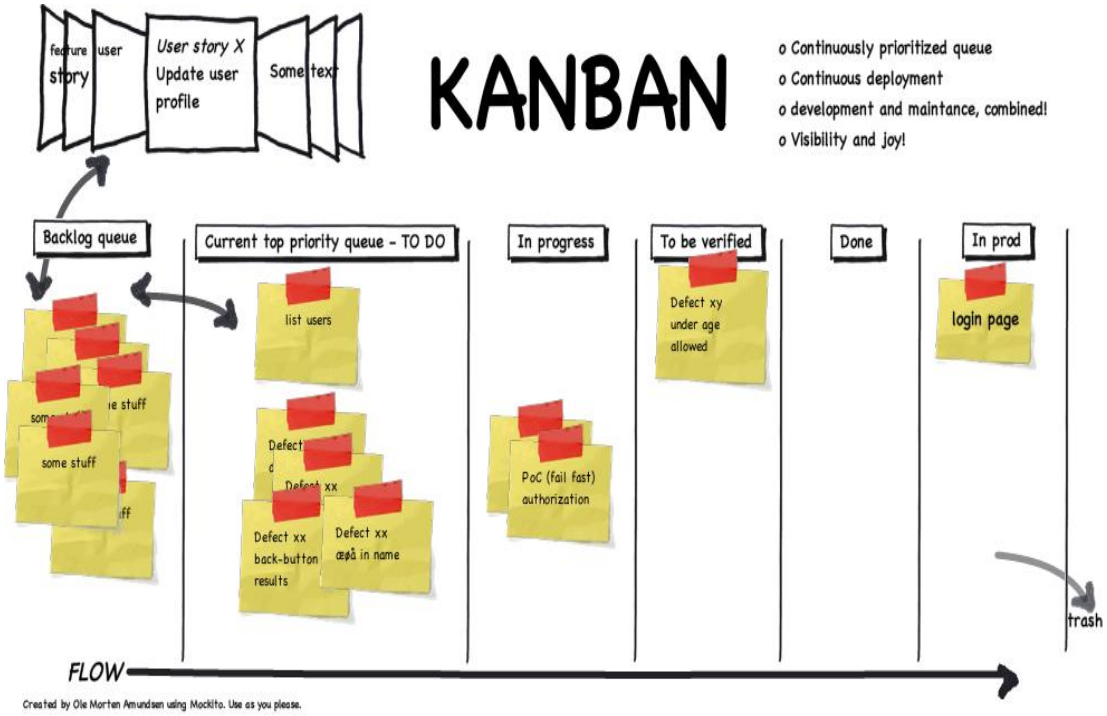
Esta metodología como se expreso anteriormente también tiene en cuenta cada uno de los principios de las metodologías ágiles en especial de SCRUM, pero a diferencia de esta ultima Kanban se encarga de darle un mayor protagonismo a la experimentación y a la mejora continua.

Es a través de esta que podemos identificar como es la visión de las diferentes perspectivas cual es el flujo de trabajo para tomar una determinación de voy a desglosar el proyecto en diferentes componentes para al finalizar estar todos convergiendo para el mismo lugar. No solamente tiene esta visión, sino también que esta siempre limitando el trabajo en curso o que es mejor conocido

como Work in Progress y el Mide el Lead Time que es quien se encarga de generar una optimización para de esta manera el Lead Time tienda a ser mucho más pequeño.

A grandes rasgos el funcionamiento de esta metodología ágil está basado en entender cuál es el flujo de trabajo que se tiene con respecto a un proyecto en especifico, con el fin de dividir el trabajo en bloques los cuales van a ser plasmados sobre el esquema que presenta la **figura 10** y así se esquematiza que cada uno de estos elementos se encuentre enfocado en el flujo de trabajo, posterior a este paso se deberá asignar el limite a cada uno de estos elementos que integran un flujo de trabajo, que como se ha mencionado anteriormente se denomina WIP (**Work in Process**) para posteriormente tener un tiempo promedio establecido para un ciclo del proceso, y de esta manera se optimizara cada momento el proceso y de esta manera reducir el tiempo de espera mucho mas.

Figura 10. Kanban for Software Development.



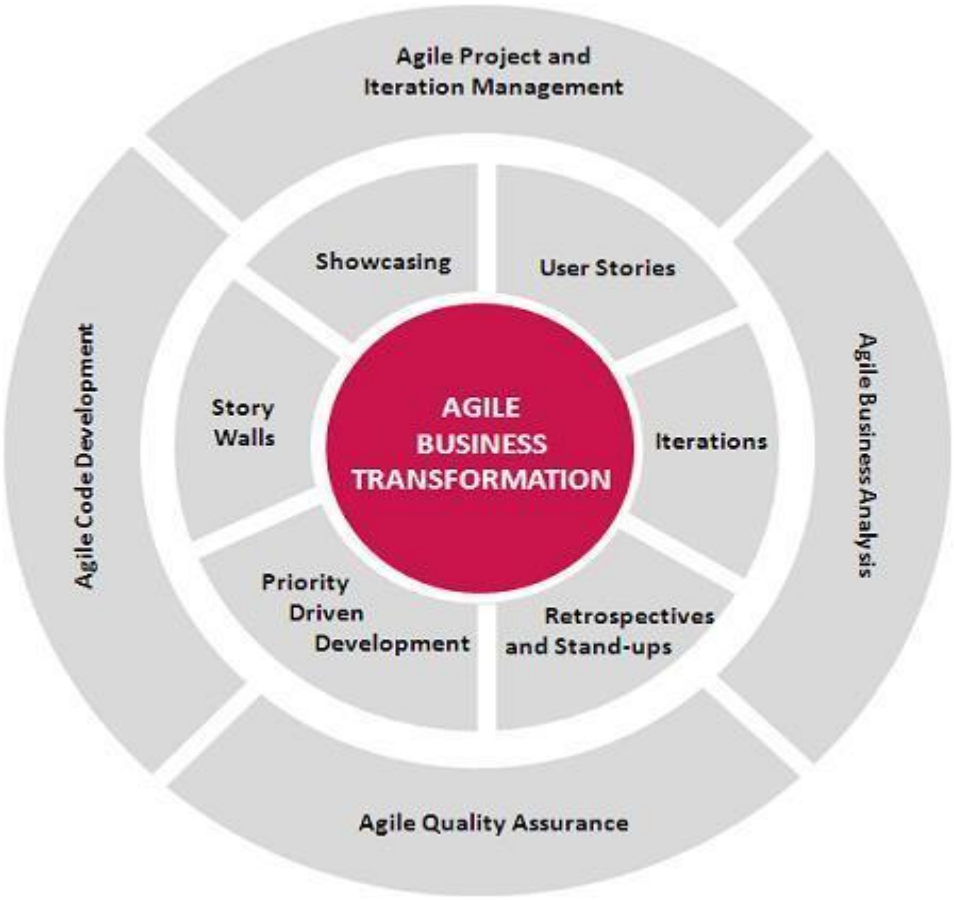
http://olemortenamundsen.wordpress.com/2010/03/19/kanban-and-scrum-combined/

Estas son tan solo algunas de las metodologías ágiles que se pueden encontrar en el mercado del desarrollo software, pues son las más representativas y en su mayoría las más utilizadas con buenos y malos resultados en proyectos de compañías grandes y pequeñas.

De un tiempo para acá se menciona acerca de una transformación ágil del negocio, esto lo que quiere indicarnos es la transformación del negocio para adoptar metodologías que permitan que

ellas funciones u operen de una manera y con una perspectiva completamente diferente, pues estas se han dado cuenta que cada día está evolucionando todo este entorno y es por esto que todas las empresas están apuntándole directamente a esta transformación para entregarles y un buen servicio y por ende ellos entreguen productos de gran calidad y que satisfagan las necesidades puntuales, además de incursionar en el nuevo camino del desarrollo ágil, por tal motivo la **figura 11** representa como es la interrelación de que en las organizaciones se lleve a cabo dicha transformación con las demás metodologías de desarrollo ágil.

Figura 11. Relación Practicas Ágiles



http://mike2.openmethodology.org/wiki/es/Transformacion_Agil_de_Negocios

4. REQUISITOS ÁGILES

Es bien sabido que los requisitos en las metodologías rígidas de desarrollo de software son un factor determinante para garantizar el éxito de los proyectos; de la misma manera juegan un papel supremo para las metodologías ágiles no obstante estas últimas no están bien clarificadas en su manera de operar e identificar los requisitos los cuales serán el pilar para el desarrollo, por ende estas últimas no poseen unos lineamientos a seguir, que al igual que las metodologías rígidas permitan un mayor éxito en cuanto a los resultados de un proyecto.

Como se menciono anteriormente la metodología XP (Extreme Programming) fundamenta la manera de obtener los requisitos del desarrollo a través de las historias de usuarios, que a grandes rasgos no es más que lo esperan los clientes que haga el sistema, mas no se centra en utilizar técnicas de elicitacion de requisitos, caso contrario sucede con la aplicación de las Metodologías Rígidas en particular en la Ingeniería de Requisitos, en donde durante su etapa de planeación usa técnicas de elicitacion como entrevistas, tormenta de ideas y priorización.

En la metodología SCRUM las principales técnicas son los backlogs, los sprints y las reuniones diarias, para el caso concreto de los requisitos en esta metodología, los backlogs del producto son la manera de recolectar los requisitos mediante una lista de backlog, tal vez esta técnica pueda compararse con un documento de requisitos que contiene información necesaria para el desarrollo, esta es plasmada desde el inicio del proyecto, dicha lista va a sufrir un proceso de priorización de cada una de las tareas plasmadas allí y que posteriormente serán distribuidas en diferentes sprints (entendiéndose un Sprint como un ciclo en el cual se desarrollaran las tareas seleccionadas), una de las políticas que tiene esta metodología es que durante la ejecución del Sprint no se podrán realizar modificaciones a los elementos allí involucrados, sino que esto podrá hacerse luego de finalizado el mismo para ser tenidas en cuenta a partir del próximo Sprint.

Al termino de cada uno de los sprints se realiza una reunión de socialización con el usuario, reunión en la cual se le va a mostrar al usuario lo desarrollado en este Sprint para que de alguna u otra manera el usuario pueda dar el aval de lo que hasta ese momento se ha trabajado, aval traducido en cuanto a funcionalidad y que cumplan con cada uno de los requisitos que debería cumplir el sistema. Dentro de esta reunión es donde el cliente pueda adicionar nuevas funcionalidades o darle a conocer al equipo desarrollador nuevos requisitos a tener en cuenta en el proyecto, requisitos que deberán ser plasmados en el Backlog del proyecto y que implicara distribuirlos en los posteriores sprints, claro que no solo se permitiría adicionar requisitos nuevos

sino también el de descartar requisitos que en dicho momento ya no sean de utilidad o necesarios para el producto final.

4.1 USUARIOS EN LOS REQUISITOS ÁGILES

Tal vez el punto clave y de mayor importancia para la adquisición de requisitos en las metodologías ágiles es tener a los clientes accesibles de manera permanente o como bien podría llamarse online, de tal manera que el cliente está implicado a lo largo de todo el proceso de desarrollo.

A diferencia de las técnicas de la Ingeniería de Requisitos tradicional las cuales tienen una percepción completamente diferente del cliente, pues este es visto como un elemento que interviene en pocas fases del proyecto, más específicamente en la fase inicial y final del mismo. Esto se puede interpretar como que en lo tradicional se enfocan en recibir de alguna manera lo que el cliente desea que haga el producto final para posterior al desarrollo del producto presentarle algo que en algunos de los casos es diferente a lo que el cliente quería expresar, y que esta diferencia puede darse por diferentes factores, que pueden estar del lado del cliente o del lado del equipo de desarrollo; esto porque es posible que el cliente no haya expresado de la mejor manera cada uno de los requisitos que debía cumplir el sistema, o caso contrario que el equipo desarrollador no haya interpretado de la mejor manera lo que el cliente estaba expresando y al final cada uno remo para un lado diferente, lo implicara establecer nuevamente un ciclo de trabajo en el cual se lleven a cabo cada una de las correcciones o modificaciones que se requieran dado que no se llenaron las expectativas del cliente.

Todo esto se da debido a no incluir o involucrar al cliente en cada una de las fases del desarrollo del producto, pues es como desentendernos del mismo y encontrarnos con ellos nuevamente tan solo para finalizar el proyecto como tal y se estaría corriendo el riesgo de trabajar en algo que no se espera.

Caso contrario pasaría en las metodologías ágiles dado que en estas el cliente hace parte del equipo de desarrollo y permitirá ir identificando posibles desviaciones del proyecto, que estén dadas por cualquiera de los casos expuestos anteriormente como el de una mala expresión del cliente o una mala interpretación por parte del equipo desarrollador, en estas metodologías adoptan al cliente como parte de ellos y lo involucran en cada una de las fases o iteraciones de la metodología con el fin de ir más firmes a cumplir con el objetivo del proyecto y de darle una satisfacción al cliente con respecto a lo entregado.

No se puede decir que involucrando al cliente de principio a fin se erradiquen por completo los inconvenientes en el desarrollo del producto, pero si se van a disminuir elementos importantes que hacen más compleja la marcha del proyecto, dichos elementos son la brecha que pueda existir

entre el equipo de desarrollo y el cliente, además de disminuir la materialización de posibles riesgos que puedan afectar el buen desarrollo del proyecto y de una manera ágil como su mismo nombre lo indica.

4.2 PRINCIPALES TÉCNICAS DE INGENIERÍA DE REQUISITOS EN LAS METODOLOGÍAS ÁGILES

Además de la interacción fundamental que deben tener los clientes/usuario en las metodologías ágiles e ingeniería de requisitos, como factor determinante para el éxito en el desarrollo de los proyectos existen también ciertas técnicas que se adaptan a los métodos ágiles como veremos a continuación:

4.2.1 Entrevistas. Esta es la principal técnica de la ingeniería de requisitos pues es de gran conocimiento y de la más utilizada dentro de las metodologías de desarrollo ágil, pues es la que brinda un gran acercamiento con los clientes/usuarios para lograr el cometido con estas. De igual manera se utiliza en todas las metodologías ágiles, para estas la conversación directa con el cliente es el método más eficaz para elicitar u obtener la información suficiente que servirá de base o soporte para lograr los objetivos trazados y brindarle una satisfacción al cliente, de tal manera que se pueda llegar a un grado de entendimiento del problema que reduzca las inferencias o malinterpretaciones del negocio por parte del equipo de desarrollo.

4.2.2 Priorización. Al igual que las entrevistas este es un método común para todas las metodologías ágiles a tal punto de que se implementan primero las características más relevantes que posee el sistema, en ciertas etapas puede ser dinámico el esquema de priorización dado que en algunos momentos pueden surgir cambios que serán tenidos en cuenta en la lista de requisitos inicial pero que como en ese inicio deben sufrir un esquema de priorización para estructurar de nuevo el esquema de trabajo a partir del momento de la inclusión de los nuevos requisitos al sistema.

4.2.3 Reuniones de Análisis. Se usa en algunas de las metodologías ágiles a lo largo de todo el proceso de desarrollo del producto, es decir que es transversal a este ultimo generando una retroalimentación constante de lo que se va trabajando, técnica la cual consiste en sostener reuniones que permitirán realizar análisis de elementos que hayan surgido en la etapa que transcurre, cada una de estas reuniones se documentan con el fin de que puedan servir de apoyo en reuniones posteriores, también de algún modo sirven para involucrar a todas las personas implicadas en el desarrollo proyecto.

4.2.4 Modelado. Es una técnica que se utiliza en las metodologías ágiles pero de una manera particular, pues es en estas que se utilizan con el fin de comunicar o darle a conocer de manera desagregadas cada una de las partes del sistema que se va a desarrollar, y caso contrario a como

pasa en las metodologías tradicionales estos elementos no llegan a formar parte de la documentación formal del proyecto ya que son de uso especifico o de momento.

4.2.5 Documentación. Para las metodologías ágiles llevar a cabo un proceso de documentación en cada una de las fases como es llevada en las metodologías tradicionales, las cuales son muy rigurosas y compleja, es ciertamente inviable de acuerdo a los principios que se encuentran plasmados en lo que se denomina el *Manifiesto Ágil*, lo que las metodologías ágiles recomiendan o en algunos casos utilizan es generar la documentación a través de los requisitos (Documento de Requisitos), en cuanto al tema de la extensibilidad se deja a consideración de cada uno de los integrantes del equipo de desarrollo o en su defecto del equipo en general, adicional a esto se tiene conocimiento que esta documentación no es muy detallado dado que se encuentra trabajando una metodología ágil.

4.2.6 Validación. Dentro del esquema de las metodologías ágiles se programan reuniones en las cuales se realice revisión y aceptación, reuniones en las cuales se pueda realizar una revisión de lo que son los objetivos y los plazos trazados para ver cómo es su grafica de cumplimiento con respecto al cronograma pactado inicialmente, es una técnica en la cual se le brinda la oportunidad al usuario de que tenga una gran participación en todo el proceso de validación del producto software en desarrollo, y que en la gran mayoría de los casos se tiene un proceso demasiado rápido el cual permitirá de alguna manera tener un retorno a la inversión más inmediato de lo pensado, y es por tanto una técnica basada en un esquema de simulación y pruebas de manera automatizadas.

4.2.7 Gestión. Dentro de cualquier proyecto, independientemente que sea de tecnología o no, se debe llevar a cabo una gestión que en muchos de los casos puede estarse más orientada a los cambios aunque no siempre está dado de esta manera, pues la gestión no debe estar enfocada solo a lo que conciernen los cambios al desarrollo del sistema sino también en ir llevando de la mano a cada uno de los integrantes del equipo de desarrollo o personas que estén involucradas, que afecten el buen desarrollo del mismo o en su defecto que se vean afectadas por cualquier cambio que allí surja, y es algo que en los proyectos de desarrollo de software se tiene en cuenta cuando se está trabajando con una metodología tradicional, pues allí se le brinda una gran importancia a la documentación y en caso de afectar el cronograma se podrá ajustar. Esto es algo que no puede suceder en las metodologías ágiles, por eso en estas últimas la gestión se realiza apoyándose demasiado en lo que se denominan tarjetas (Historias de Usuarios), backlog o listas de características (esto dependiendo del nombramiento en cada una de las metodologías).

Es por esto que muchos de los críticos o expertos en el tema infieren que el control de cambios no genera ningún beneficio económico para el proyecto dado que por el contrario implica mayor esfuerzo en la generación de la documentación, pero como se expreso anteriormente si es algo que está más enfocado en las metodologías tradicionales. Si pueden contener una serie de diferencias en cuanto a la manera en como es el nivel de detalle con el que se estructuran en cada

una de las metodologías (Tradicional o Ágil), dado que en las ágiles se estipula se vayan completando una vez van sucediendo iteraciones posteriores.

4.2.8 Requisitos No Funcionales. Con este tipo de requisitos, se tiene una gran dificultad en su definición cuando se está utilizando una metodología ágil, por lo tanto no se tiene estructurado un manejo adecuado para estos. En la mayoría de los casos los clientes no los tienen muy presentes, pero van a ir surgiendo a medida que va transcurriendo el desarrollo del sistema.

4.3 HISTORIAS DE USUARIO

Las historias de usuario representan el tramo el cual conlleva por un camino, es decir describen cada uno de los eventos que encaminara al equipo de desarrollo a alcanzar un objetivo y que se considera son quizás la técnica más utilizada en las metodologías ágiles todo con el fin de brindarle un tratamiento a los requisitos, sin embargo esta no es contemplada de tal forma en la Ingeniería de Requisitos tradicional.

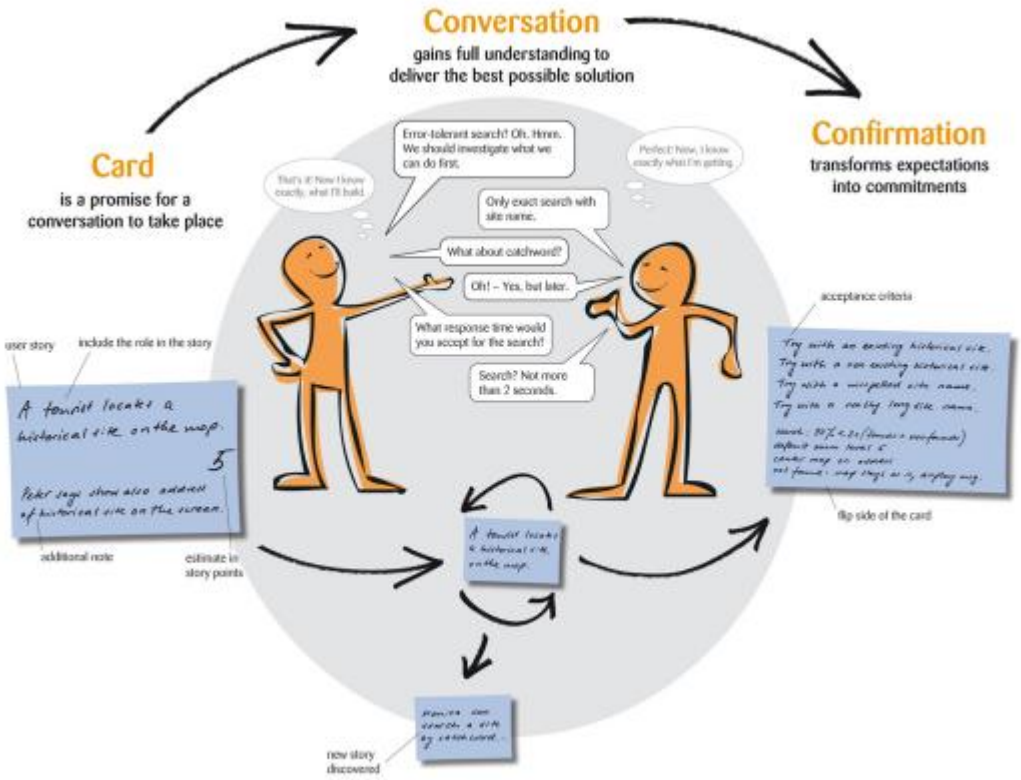
Estas a diferencia del documento de requisitos en las metodologías tradicionales, son expresadas en lenguaje unipersonal, es decir en palabras del usuario pero tienen el mismo propósito de los casos de uso, propósito que está enfocado a brindar una buena estimación de tiempos cuando a la planificación del proyecto nos referimos, y en reemplazar el documento de requisitos.

Cuando se habla que son expresadas en lenguaje unipersonal o palabras del usuario, es dado que son los clientes quienes tienen la tarea específica de construir las historias, donde ellos plasmaran cada una de las cosas que el sistema debería hacer, excluyendo de allí toda la terminología técnica, y la principal diferencia con respecto a la Ingeniería de Requisitos es el nivel de detalle trabajado en cada una, pues es en las historias de usuario que se debe proporcionar el suficiente detalle que le permita al equipo de desarrollo obtener una estimación de los costes en la implementación de las mismas.

Es luego de esto que el equipo de desarrollo interactuara de una manera más especifica con el usuario, interacción que le brindara un mayor detalle de la descripción brindada de cada uno de los requisitos plasmados en la historia de usuario, sin embargo esto podría implicar de alguna u otra manera un mayor esfuerzo del que se había estimado anteriormente.

Para una mejor visión de lo que debe expresar una Historia de Usuario, la **figura 12** estará representando de una manera más grafica su composición.

Figura 12. Historia de Usuario



http://www.comp.lancs.ac.uk/~gacitur1/AREW11/documents/Working%20with%20User%20Stories.pdf

Teniendo en cuenta lo descrito en cada uno de los anteriores renglones, se da claridad de que cada una de las fases que se encuentran de alguna u otra manera establecidas dentro de las metodologías tradicionales, se encuentran también incorporadas dentro de las metodologías ágiles con la diferencia que en estas últimas no se existe una separación entre todas las fases, pues dentro de cada iteración se llevan cabo, algo que es completamente opuesto dentro de las metodologías tradicionales ya que allí se encuentran demasiado marcadas para trabajar de una manera secuencial.

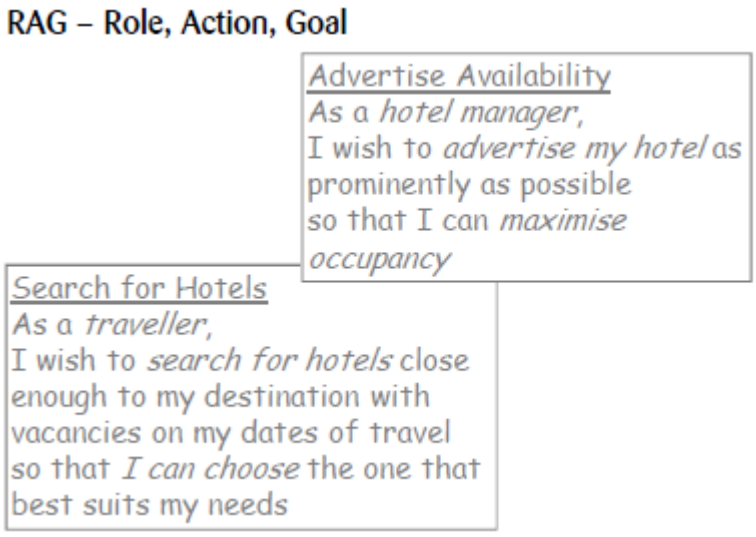
Algunas de las ventajas que puedan tener las metodologías ágiles, son básicamente que todo el esfuerzo de la fase de requisitos la sostiene hasta el momento de dar inicio a la implementación, sin embargo para las metodologías tradicionales la documentación está establecida como una manera singular de compartir el conocimiento entre cada uno de los participantes del proyecto, permite reducir de alguna manera el riesgo de la pérdida del conocimiento por causa de la exclusión de un miembro del equipo, lo cual algunos de los expertos en el tema dan a entender que esto puede verse como inconveniente para el buen desarrollo de sistemas a través de las

metodologías ágiles. Algo que sucede de manera diferente en estas últimas, pues es en estas que la documentación se ve con una orientación más hacia el cliente, más compacta y que pueda brindar una facilidad en cuanto al manejo o utilización de la misma, por tal motivo es que al reducir el tiempo en la generación de una documentación muy extensa o detallada les va a brindar una mayor productividad.

Luego de tener una contextualización tan detallada como la anterior se concluye que las metodologías ágiles poseen una serie de ventajas en referencia a todo el desarrollo de sistemas software; dichas ventajas están enfocadas a la interacción con el cliente y en mejorar sustancialmente la eficiencia del desarrollo en dos aspectos de gran relevancia como lo son el tiempo y costos. Claro está, que se deben dejar ciertos elementos en claro, pues estas metodologías aun tienen un nivel de madurez en proceso, en especial en el ámbito de la Ingeniería de Requisitos, dado que no tienen una estructuración afianzada a nivel de la relación con dicha ingeniería a través del proceso riguroso que se debe llevar a cabo en esta área, que permita obtener los mayores beneficios para todas las partes implicadas, y que den esas bases solidas que se necesitan para el buen inicio del desarrollo del proyecto, pues es sobre estas bases que se va a soportar la exitosa ejecución del proyecto.

Cuando se centra un precedente acerca de las historias de usuario estamos hablando de un elemento de gran importancia para algunas de las metodologías ágiles, estas poseen elementos cruciales para de alguna u otra manera ser catalogadas como su nombre lo indica, es por esto que la **figura 13** nos muestra a simple vista lo que debe contener una historia de usuario para ser catalogada como tal.

Figura 13. Elementos Historias de Usuario



http://www.comp.lancs.ac.uk/~gacitur1/AREW11/documents/Working%20with%20User%20Stories.pdf

4.4 PERSPECTIVA DESDE LA INGENIERÍA DE REQUISITOS PARA LAS METODOLOGÍAS ÁGILES

Desde el punto de vista de la Ingeniería de Requisitos muchas personas ven en las Metodologías Ágiles la justificación más oportuna para tirar a la borda todo lo que se proclama en dicha ingeniería, además al no hacer un buen uso o implementación de la Ingeniería de Requisitos conllevara a que cada uno de los proyectos tenga una gran carencia de buen calidad, lo cual es un aspecto demasiado importante para los clientes.

A menudo las fallas de los proyectos se encuentran en la falta de atención oportuna en los procesos de requisitos, dado que este es el punto que cimienta las bases del proyecto se convierte en la palabra clave a la hora de hablar del éxito del mismo.

Una de las principales justificaciones para no hacer el correcto uso en las técnicas de requisitos es la falta de tiempo, ya que todas las compañías están enfocadas en brindarle una satisfacción enorme a cada una de las necesidades que tienen sus clientes pero son estos últimos los que están indicando el momento para recibir su producto que están enmarcado en el instante, además del deseo de tener al cliente feliz entregando productos tan pronto como sea posible. Es por estas y muchas otras razones por las cuales las compañías desarrolladoras han mostrado gran interés en adoptar metodologías ágiles.

Para los más ortodoxos de la Ingeniera de Requisitos, el proceso de requisitos que realizan las metodologías ágiles no es el más adecuado y de alguna manera no se encuentra con el enfoque correcto, ya que requieren de mas ciclos de validación que los necesarios y además más grave aún confían demasiado en las personas, dado que dentro de un proyecto es demasiado riesgoso sopesar las responsabilidades en las personas ya que en cualquier momento alguna de estas puede partir o puede sufrir algún inconveniente y cuando una personas parte, se va junto con el conocimiento y en las metodologías ágiles esto puede verse con mayor frecuencia al ver que el tema de la documentación está un poco relegada, lo que implicaría que la persona que asuma en este espacio no tendrá una facilidad para adaptarse al trabajo que se venía llevando a cabo.

No obstante hay muchas apreciaciones acerca de la alta productividad que tienen las metodologías ágiles, pero lo que no está muy evidenciado es cuales son realmente las causas de dicha productividad, para algunos podría deberse a la conformación de los equipos de trabajo, conformado por las personas más brillantes y con altas habilidades para programar o también la contratación de un excelente consultor que brinde apoyo a todo el proceso de desarrollo.

Existen cuatro buenas prácticas de la Ingeniería de Requisitos que podrían enriquecer y aportarle mayor calidad a las metodologías ágiles como lo son la interacción con el cliente, análisis (Validación y Verificación), los requisitos no funcionales y la gestión del cambio.

4.4.1 Interacción con el Cliente. La importancia de esta práctica para el proceso de desarrollo de software ha sido aprobada y generalizada por las metodologías existentes, a tal punto que se convierte en la técnica clave para el éxito de los proyectos por lo tanto en la Ingeniería de Requisitos se le ha dado suma importancia para mejorar esta interacción, todo a través de la creación de diferentes técnicas que permitan enriquecer este aspecto y así lograr que el proyecto en curso tenga un mejor desempeño que los dirija a lograr los objetivos propuestos.

Dentro de este marco de la interacción con el cliente, se evidencia que existen dos tipos de clientes que dentro de las metodologías ágiles no se ven tan marcados, pero que en estas últimas debería de tenerse muy presente dado que es la mejor manera de obtener una muy buena interacción entre los involucrados en el proyecto. Los tipos de clientes son el *usuario* que se da como la persona que esta interactuando día a día de forma directa con el sistema, aunque también está el *cliente* que es como tal quien tiene las necesidades a suplir mediante el desarrollo de software y al igual que los desarrolladores están muy interesados en el proceso de construcción.

El trabajo en esta área está centrado en cuatro ciencias de interacción con individuos, lo cual permite realizar una mejor elicitacion de requisitos:

- Psicología Cognitiva, esta permite entender las dificultades que puedan tener las personas involucradas para formular lo que desean que haga el sistema.
- Antropología, ayuda a observar a las personas involucradas para instarlas a averiguar cómo podría el sistema mejorar su desempeño.
- Sociología, permite generar una buena interacción entre todas las personas involucradas.
- Lingüística, ayuda a propiciar una buena comunicación de tal forma que se puedan expresar claramente las ideas y necesidades por parte de las personas involucradas.

Un aspecto muy importante es identificar muy bien a todos los individuos que participan en este proceso, de tal manera que sepamos distinguir los que tienen voz y voto y los que están interesados (usuarios) para determinar cuáles son las necesidades de cada uno y de esta forma tener una buena información para el análisis de los requisitos.

Las metodologías ágiles de acuerdo a sus principios tienen este enfoque de interacción con el cliente muy bien fundamentado pero podrían estar incurriendo en un error al asumir que los clientes ya saben todo lo que tienen que saber, lo cual en la mayoría de los casos no es realmente cierto, para este caso en particular se debería seleccionar muy bien a los clientes adecuados ya que harán parte del proyecto a lo largo del proceso de desarrollo.

4.4.2 Análisis. En esta etapa se hace un estudio de lo que se ha definido previamente a nivel del modelo, con el fin de encontrar deficiencias que permitan corregir y avanzar en los objetivos que se han trazado. Para esto es indispensable hacer uso de los siguientes conceptos:

- **Validación.** En términos generales se entiende como las pruebas que se deben realizar para identificar posibles errores y su oportuna corrección, del mismo modo es utilizada para encontrar nuevas necesidades. En otras palabras la validación lo que busca es determinar si cumple con las necesidades hechas por el usuario.

- **Verificación.** Este aspecto está enfocado en determinar a manera de control si efectivamente se está cumpliendo con los parámetros establecidos para la realización. En otras palabras la verificación busca comprobar si se está construyendo de forma correcta el producto. En general estas metodologías no emplean los aspectos de verificación.

La aplicación de formalismos en Ingeniería de Requisitos son en gran medida adoptados por la ingeniería de software, los cuales deberían estar un poco mas introducidos en las metodologías ágiles para proporcionar una infraestructura básica para el análisis, por tanto deberían incluir la verificación como proceso de vital importancia ayudando a mejorar la calidad del desarrollo.

4.4.3 Requisitos no Funcionales. En este aspecto las metodologías ágiles no se concentran demasiado ya que para estas los requisitos no funcionales van a ir apareciendo a medida que va avanzando el proyecto. Desde el punto de vista de Ingeniería de Requisitos estos se deberían de tener en cuenta al mismo tiempo que las historias de usuario pues son consideradas de lado de los clientes.

4.4.4 Gestión del Cambio. Este es un aspecto inherente a los procesos de desarrollo de software de tal manera que al surgimiento de eventuales cambios se pueda llevar a cabo determinadas acciones que me impliquen el menor impacto para el proyecto.

Cuando se habla de gestión de requisitos se está hablando de una condición previa a todo lo que implica esa gestión del cambio, en donde esta última se basa en la identificación, la organización y el control de cada uno de los cambios que puedan surgir a lo largo del proceso del desarrollo del software, algo muy similar sucede con la gestión de requisitos, dado que con esta gestión se pretende identificar, priorizar y controlar los cambios en los requisitos del sistema, para tener presente que puede afectar el buen desarrollo de la construcción, pues estos cambios irán apareciendo muy probablemente en cada una de las iteraciones del proyecto.

Las metodologías ágiles no deben darse el lujo de no contar con la gestión de requisitos ya que esta les puede proporcionar el esquema de la rastreabilidad del proyecto, entendiendo como rastreabilidad todo el esquema de seguimiento en cada una de las fases del proceso de desarrollo del software, pues en las metodologías tradicionales se plasma dicha rastreabilidad a través de procedimientos que estructuran en documentación que permitirá conocer cuál es el histórico, la

ubicación y la trayectoria que ha tenido el desarrollo del software a lo largo del proceso, y así de alguna u otra manera tener claridad de cuáles son los cambios surgidos y efectuados para que el conocimiento en algún momento pueda perderse.

Es por todo esto que no se trata solo de generar documentos breves como en la metodología ágil XP en la cual su punto de referencia son las historias de usuario, pues es complejo tener este elemento como parte fundamental en lo que se denomina la gestión de requisitos ya que son elementos que de alguna u otra manera no expresan de manera clara cuál ha sido la rastreabilidad que ha tenido cada uno de los requisitos elicitados desde el inicio y que en cualquier momento del tiempo en el proyecto surgirá un cambio que podría afectar directa o indirectamente otros requisitos o al proyecto global.

Cada uno de estos elementos que se describen son de vital importancia dentro de todo el proceso de la ingeniería de requisitos y que de alguna u otra manera no son bien tenidos en cuenta por las metodologías ágiles.

CONCLUSIONES

La realización de este trabajo nos ha sido de suma importancia para nuestra formación profesional ya que con este damos un vistazo a partir del Proceso de Ingeniería de Software la cual entre otras cosas es nuestra línea de énfasis y nos apasiona el tema que dio pie para la realización de este trabajo.

Dentro de dicha Ingeniería de Software nos encontramos con tal vez la parte más difícil y más importante para el éxito de los proyectos de software en el mundo actual como lo es la Ingeniería de Requisitos, la cual con un correcto uso o aplicación va determinando la exitosa vida del Proceso de Desarrollo. Generando un inmenso valor para las compañías tanto desarrolladoras como las que requieren del producto para satisfacer las necesidades que se presentan en los negocios.

Por otro lado y no menos importante se encuentran las Metodologías Ágiles que como ya se había mencionado anteriormente son una corriente que día a día toma mucha fuerza y que posiblemente a futuro tomara muchísima más importancia de la que hoy en día goza, por esta razón es que nos hemos inclinado a indagar mucho más en el tema que nos permitiese generar más conocimiento no solo para nosotros sino para las personas interesadas en el tema.

Hemos logrado mediante este trabajo integrar dos conceptos importantes como lo es la Ingeniería de Requisitos orientada a las Metodologías Ágiles como una forma de robustecer estas últimas y de esta manera ayudar en la consecución de mayor calidad en los productos a desarrollar.

BIBLIOGRAFIA

Ayerbe Bernal, Rafael. Paper “Ingeniería de Requisitos en los Métodos de Desarrollo Ágiles”, 2007.

Eberlein, Armin y Sampaio do Prado Leite, Julio Cesar. Paper “Agil requirements Definition: A view from Requirements Engineering”

Kruchten, Philippe. The Rational Unified Process an Introduction. 2003.

Leffingwell, Dean. Agile Software Requirements – Lean Requirements Practices for Teams, Programs, and the Enterprise. 2011.

Lee, Christopher. Guadagno, Luigi y Jia, Xiaoping. An Agil Approach to Capturing Requirements and Traceability.

http://www.computer.org/portal/web/swebok/html/ch2

http://www.willydev.net/descargas/prev/TodoAgil.Pdf

Leonardi, Maria Carmen y Sampaio do Prado Leite, Julio Cesar. Un Proceso para XP Basado en Reglas de Negocio. De
http://wer.inf.puc-rio.br/wer01/Eli-Req-2.pdf

http://searchsoftwarequality.techtarget.com/feature/Collaboration-in-Agile-development-Requirements-analysis-is-a-team-effort

http://www.seccperu.org/files/Metodologias%20Ágiles.pdf

http://materias.fi.uba.ar/7500/schenone-tesisdegradoingenieriainformatica.pdf

http://interfaces-siges.googlecode.com/files/Metodologias%20Ágiles.doc

http://www.slideshare.net/JuanGomez13/metodologas-ágiles-de-desarrollo-de-software

http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_requisitos

http://www.cs.utexas.edu/users/downing/papers/Agile2007.pdf

http://www.latindex.ucr.ac.cr/intersedes10/10-art_11.pdf

http://www.rodolfoquispe.org/blog/que-es-la-ingenieria-de-requisitos.php

http://www.slideshare.net/ssharLudena/ingeniera-de-requisitos

Martínez, Alejandro y Martínez, Raúl. Guía a Rational Unified Process. De http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Guia%20RUP.pdf

http://www.itenea.com/file.php/1/Metodologias_Ágiles_de_Gestion_Kanban.pdf

http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software#Etapas_del_proceso

http://www.proyectalis.com/documentos/KanbanVsScrum_Castellano_FINAL-printed.pdf

http://olemortenamundsen.wordpress.com/2010/03/19/kanban-and-scrum-combined/