

**FORMULACIÓN DE UNA METODOLOGIA PARA LA GESTIÓN DE PROYECTOS
DE DESARROLLO DE SOFTWARE, A TRAVÉS DE PRINCIPIOS Y PRÁCTICAS
ÁGILES CON ENFOQUE PMI**

AUTOR

Estephani Londoño Hoyos

Ingeniera de Sistemas y Computación – Universidad del Quindío
stefylondono@gmail.com

**“Artículo presentado como trabajo final de Especialización en Gerencia Integral de
Proyectos”**

TUTOR

Ing. Guillermo Roa Rodríguez, MSc

Maestría en Ingeniería Mecatrónica de la Universidad Nueva Granada
Especialización en Gerencia de proyectos de la Universidad Nueva Granada
Ingeniero en Mecatrónica - Universidad Militar Nueva Granada
Coordinador Especialización en Gerencia Integral de Proyectos y
Maestría en Gerencia de Proyectos de la Universidad Militar Nueva Granada
guillermo.roa@unimilitar.edu.co



**UNIVERSIDAD MILITAR NUEVA GRANADA
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN EN GERENCIA INTEGRAL DE PROYECTOS
DICIEMBRE 2015**

FORMULACIÓN DE UNA METODOLOGIA PARA LA GESTIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE, A TRAVÉS DE PRINCIPIOS Y PRÁCTICAS ÁGILES CON ENFOQUE PMI

FORMULATION OF A METHODOLOGY FOR PROJECT MANAGEMENT SOFTWARE DEVELOPMENT, THROUGH PRINCIPLES AND PRACTICES AGILE WITH FOCUS PMI

Estephani Londoño Hoyos

Estudiante Especialización en Gerencia Integral de Proyectos Universidad Militar Nueva Granada.

Ingeniera de Sistemas y Computación Universidad del Quindío.

Auditora Interna de Aseguramiento de Calidad en Heinsohn Business Technology.

Bogotá, Colombia

stefylondono@gmail.com

RESUMEN

El presente trabajo tiene como propósito la formulación de una metodología para la gestión de proyectos de desarrollo de software, a través de principios y prácticas ágiles con enfoque PMI, que podrá ser implantada en la empresa Heinsohn Business Technology como una solución a las dificultades actuales para la gestión de proyectos de corta duración, presupuestos reducidos y en los cuales los clientes exigen resultados rápidos y funcionales, adicionalmente se presenta como requisito para culminar el proceso educativo y obtener el grado de especialista en Gerencia Integral de Proyectos de la Universidad Militar Nueva Granada. El documento presenta, en primer lugar las dificultades de las técnicas tradicionales para la gestión de proyectos y de estas se determinan una serie de requerimientos. Posteriormente se muestra la investigación realizada en cuanto a metodologías ágiles de las cuales se enfatizó en las metodologías Scrum y Extreme Programming (XP) después de realizar una valoración teniendo en cuenta fuentes y calidad de información disponible, de dicha investigación se logró identificar complementariedad entre las metodologías caracterizadas y las áreas de conocimientos del PMBOK. Finalmente a través de la ejecución del documento presentado, fue posible describir las actividades de la metodología propuesta y presentar el diseño de la formulación del proceso para la gestión de proyectos de desarrollo de software.

Palabras Clave: Metodología Ágil, Manifiesto Ágil, Scrum, Extreme Programming, requerimientos.

ABSTRACT

This work aims at the development of a methodology for managing software development projects through agile principles and practices PMI approach that can be implemented in the company Heinsohn Business Technology as a solution to the current difficulties Project management will short, tight budgets and where customers demand fast and functional results additionally presented as a requirement to complete the educational process and obtain the degree of specialist in Integral Project Management New Granada Military University. The paper presents, first the difficulties of traditional techniques for managing projects and a number of these requirements are determined. Subsequently, the research shows in terms of agile methodologies which was emphasized in the Scrum and Extreme Programming (XP) methodology after performing an assessment taking into account sources and quality of information available, the investigation was able to identify complementarities between characterized methodologies and PMBOK knowledge areas. Finally through the implementation of the presented document it was possible to describe the activities of the proposed methodology and present the design of the formulation process for managing software development projects.

Keywords: Methodology Agile, Manifest Agile, Scrum, Extreme Programming, requirements.

INTRODUCCIÓN

El **Project Management Institute (PMI)** establece la gestión de proyectos como la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto para satisfacer sus requisitos. Para que la gestión de proyectos sea satisfactoria es indispensable partir de una planificación estructurada y exhaustiva que permita alcanzar los objetivos del proyecto optimizando la asignación de tiempos, recursos y costos pero manteniendo la calidad. El producto producido por la PMI con mayor reconocimiento a nivel mundial es el **Project Management Body of Knowledge (PMBOK)**, como su nombre lo indica es un conjunto de conocimientos y prácticas aplicables a cualquier situación que requiera la formular y ejecución de un proyecto; es una guía de estándares internacionales para que los profesionales puedan adaptar a cada caso y contexto particular los procesos adoptando las mejores prácticas. La importancia del PMBOK es que provee un marco de referencia formal para desarrollar proyectos, guiando y orientando a los gerentes de proyectos sobre la forma de avanzar en los procesos y pasos necesarios para la construcción de resultados y alcanzar los objetivos. El PMBOK documenta la información necesaria de los procesos generales de la administración de proyectos que son: iniciación, planificación, ejecución, supervisión, control y cierre de un proyecto. [1]

La investigación de mercado del PMI muestra que expertos en dirección de proyectos están promoviendo los principios y las prácticas ágiles como una técnica para gestionar proyectos de manera exitosa. Encuentran valor en las prácticas y principios ágiles para entregar proyectos más rápido, con menos desperdicio y menos costo debido a malos entendidos o a requisitos pobremente definidos. Las prácticas y los principios ágiles son temas que tienen cada vez más importancia en la profesión de la dirección de proyectos. Quienes practican la dirección de proyectos los pueden usar para gestionar el cambio con éxito, mejorar las comunicaciones, bajar los costos, aumentar la eficiencia y demostrarles valor a los clientes e interesados. La filosofía ágil usa modelos de organización basados en la gente, en la colaboración y en los valores compartidos. El Manifiesto de ágil presenta los principios de dicha filosofía. La misma usa la planificación iterativa con entregas incrementales; una respuesta flexible y rápida ante los cambios; y una comunicación abierta entre los equipos, los interesados y los clientes. [1]

Debido a las necesidades cambiantes del negocio, la competitividad de la industria y la importancia de reducir tiempo y costos en los proyectos software sin sacrificar la calidad de los mismos y logrando por ende la fidelización y recomendación de los clientes surgen diversas metodologías ágiles que proporcionan marcos de referencia y procesos cuyas prácticas sustentan los principios del Manifiesto de ágil. El manifiesto ágil es un documento que resume en cuatro valores y doce principios las mejores prácticas para el desarrollo de software, promueve el desarrollo más rápido conservando la calidad. [3]

Las metodologías ágiles son una serie de técnicas para la gestión de proyectos que han surgido en contraposición a los métodos clásicos de gestión, todas las

metodologías que se consideran ágiles cumplen con el manifiesto ágil. Entre las metodologías ágiles más usadas se encuentran, **SCRUM** que es un marco de trabajo que nos proporciona una serie de herramientas y roles para, de una forma iterativa, poder ver el progreso y los resultados de un proyecto y **Extreme Programming (XP)** que es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. [4]

Heinsohn Business Technology es compañía colombiana especializada en desarrollo e implementación de soluciones software actualmente cuenta con una metodología de desarrollo de software en cascada que es un enfoque de dirección de proyectos orientado a las fases secuenciales donde cada fase se debe planificar y completar antes de poder avanzar a la siguiente. De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos y tiempos del desarrollo. En la vida real, un proyecto rara vez sigue una secuencia lineal, y es por esta razón que cada día son más los clientes que solicitan llevar a cabo proyectos de manera más ágil donde se reduzcan los tiempos de entrega, se optimicen los recursos y se minimicen los costos. Es por esta razón que la presente investigación busca identificar los principios y prácticas ágiles para la gestión de proyectos software haciendo un paralelo con la guía PMBOK y los estándares del PMI para lograr la formulación de una metodología para la gestión de proyectos de desarrollo de software.

1. MATERIALES Y MÉTODOS

1.1 Evaluación del estado actual del área de gestión de proyectos de desarrollo de software de la empresa Heinsohn Business Technology

Heinsohn Business Technology es compañía colombiana especializada en desarrollo e implementación de software empresarial, con más de 35 años de experiencia en el mercado nacional e internacional, aproximadamente 1.200 clientes emplean soluciones Heinsohn. Cuenta con valoración CMMI Nivel 5 de madurez, principal estándar de la industria desarrollado por el Software Engineering Institute (SEI) y certificación en normas internacionales de calidad como ISO 9001, 19011 y 27001.

Posee un sistema de gestión de calidad llamado HUP (Procesos Unificados Heinsohn) en el cual se encuentra documentados cada uno de los procesos y procedimientos a seguir en la compañía a nivel estratégico, misional y de apoyo; entre ellos el proceso de gestión de proyectos guiado por el PMBOK y los estándares del PMI.

Cada uno de los procesos se encuentra enmarcado dentro de un ciclo PHVA (Planear, Hacer, Verificar y Actuar).

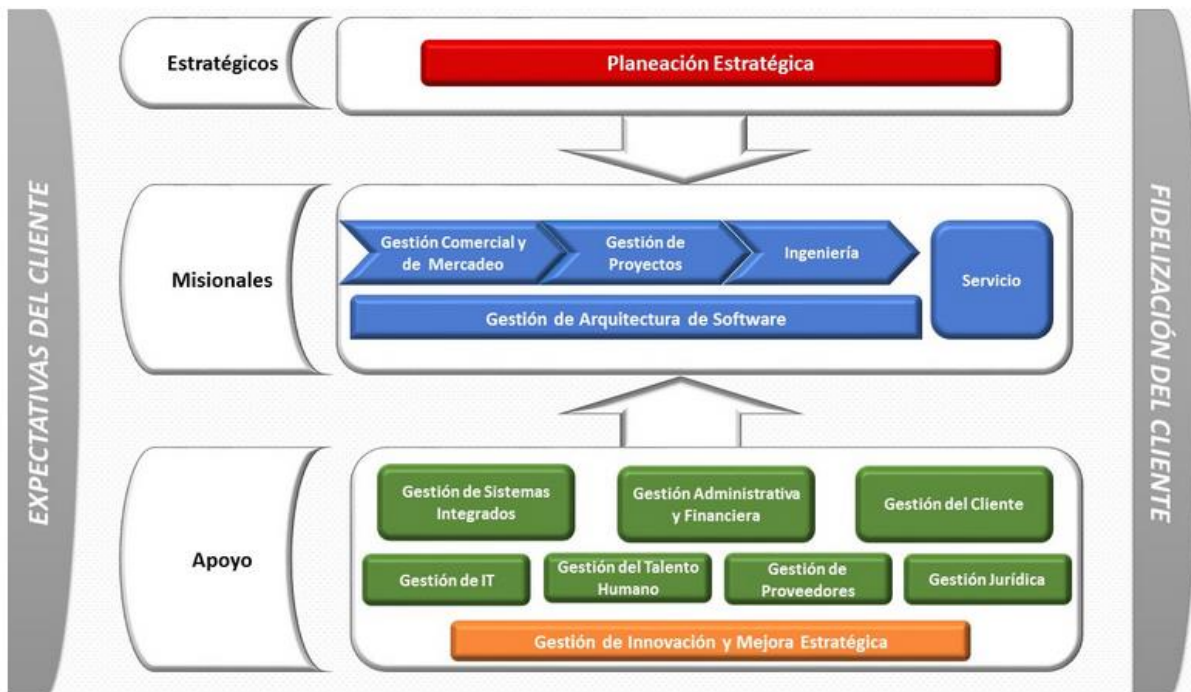


Figura 1. Mapa de procesos de la empresa Heinsohn Business Technology

Fuente: Sistema de Gestión de Calidad (HUP), 2015.

El proceso de gestión de proyectos inicia con la realización de una venta basada en una propuesta comercial y finaliza con la entrega del producto del cliente. El propósito del proceso de gestión de proyectos es definir, planear y controlar la correcta iniciación, ejecución, control y finalización de los proyectos y/o productos; organizando y administrando los recursos, de forma tal que sea terminado completamente dentro de las restricciones de alcance, tiempo y costos planteados a su inicio.

El proceso de gerencia de proyecto está compuesto por los procedimientos de Iniciar proyecto, Planear proyecto, Ejecutar proyecto, Monitorear y controlar proyecto y Cerrar proyecto. Al iniciar el proyecto se requiere realizar algunas actividades para garantizar la correcta ejecución como son la firma del contrato, el presupuesto inicial y la consecución del recurso humano del proyecto. En la planeación del proyecto se elabora el plan de proyecto que contiene la información necesaria para lograr los objetivos específicos del proyecto. Durante la ejecución del proyecto se realiza la gestión de los recursos humanos y técnicos de acuerdo a lo planeado y realizar la facturación que asegure la ejecución del presupuesto en forma correcta. [2]

Es necesario monitorear y controlar en forma permanente los riesgos, el presupuesto, la calidad, realizando las mediciones y análisis que lleven a realizar las mejoras a que haya lugar para garantizar el cumplimiento de los objetivos del proyecto. Al finalizar el proyecto se deben realizar las actividades de cierre como ejecutar las actividades jurídicas de acuerdo al contrato, documentar las lecciones aprendidas, realizar la liberación de los recursos, cerrar el proyecto en las herramientas internas y realizar el acta de cierre. [2]

1.2 Identificación de dificultades de las técnicas tradicionales de dirección de proyectos de desarrollo de software guiados por el PMBOK y los estándares del PMI en la empresa Heinsohn Business Technology

Actualmente la empresa Heinsohn Business Technology cuenta con una metodología de desarrollo de software en cascada que es un enfoque de dirección de proyectos orientado a las fases secuenciales donde cada fase se debe planificar y completar antes de poder avanzar a la siguiente. De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos y tiempos del desarrollo.

En la vida real, un proyecto rara vez sigue una secuencia lineal, y es por esta razón que cada día son más los clientes que solicitan llevar a cabo proyectos de manera más ágil. Teniendo en cuenta dicha premisa se pretenden identificar las principales dificultades de las técnicas tradicionales de dirección de proyectos software guiados por el PMBOK y los estándares del PMI, para de esta forma tener los puntos de partida para seleccionar una o más metodologías ágiles que optimicen la gestión exitosa de los proyectos software. Adicionalmente es importante generar un paralelo entre la de la guía PMBOK y los estándares del PMI con las metodologías ágiles para validar que efectivamente no se excluyen grupos de procesos ni áreas de conocimiento.

Los métodos ágiles son necesarios para la gestión de proyectos porque dependiendo el tipo de proyecto pueden variar también las formas para abordarlo. Algunos proyectos, especialmente proyectos que precisan de trabajadores con conocimientos especializados, se desarrollan en ambientes de rápida acción con tiempos limitados, por lo que es más adecuado un método ágil para desarrollarlos. A continuación se presenta un cuadro comparativo entre las características del trabajo basado en conocimientos (Construcción de Software) y trabajo industrial **Tabla 1**.

Tabla 1. Diferencias entre trabajo basado en conocimientos y trabajo industrial

<i>Características del trabajo basado en conocimientos (Construcción Software)</i>	<i>Características del trabajo Industrial</i>
El trabajo es invisible (Se produce algo intangible).	El trabajo es visible (Se produce algo tangible)
El trabajo es cambiante	El trabajo es estable
Énfasis en hacer cambiar las cosas	Énfasis en hacer avanzar las cosas
Menos estructuras con más decisiones	Más estructura con menos decisiones
Se centran en las preguntas correctas	Se centran en las repuestas correctas
Los trabajadores comprenden la tarea	Define las tareas
Da autonomía	Control y Dirección
Innovación Continua	Reglas estrictas
Se centra en la calidad	Se centra en la cantidad
Aprendizaje y formación continuos	Mide el rendimiento con reglas estrictas
Trata a los trabajadores como activos, no como gastos.	Minimiza el costo de los trabajadores para una tarea.

Fuente. Métodos Ágiles, 2011.

Las principales dificultades y limitaciones de las técnicas tradicionales de dirección de proyectos de desarrollo de software guiados por el PMBOK, los estándares del PMI y las metodologías tradicionales de desarrollo de software identificados en la empresa Heinsohn Business Technology se pueden observar en la **Tabla 2**.

Tabla 2. Principales dificultades y limitaciones de las técnicas tradicionales de dirección de proyectos de desarrollo de software

Área de Conocimiento	Limitación o Dificultad
Alcance	Se limitan la participación del cliente sólo a reuniones de control, reduciendo de manera significativa sus aportes. Lo que dificulta encontrar las verdaderas necesidades del cliente.
Alcance	Se asume que no se van a presentar cambios una vez iniciado el proyecto. Lo cual no es cierto debido a los requisitos variables del cliente.
Alcance	Excesiva documentación en la etapas inicial lo que retrasa la entrega tangible del producto al cliente.
Tiempo y Costo	Facilidad de desviarse del tiempo planificado (debidos a los intentos por cambiar el sistema para que se incluya los nuevos requisitos del sistema).
Tiempo	Largos tiempos de espera por parte del usuario para ver los resultados.
Riesgos	Impacto de los cambios alto debido a que el proyecto ya se encuentra en fases avanzadas.

1.3 Identificación de los requerimientos actuales del área de gestión de proyectos de desarrollo de software de la empresa Heinsohn Business Technology teniendo en cuenta las necesidades de la industria en cuanto a metodologías ágiles

Debido a la necesidad de la industria del software de agilizar los tiempos de entrega y los costos de desarrollo, se han venido estableciendo diversas metodologías que proponen principios y prácticas para abordar los proyectos de manera más eficiente pero sin dejar a un lado las buenas practicas propuestas por las técnicas tradicionales de dirección de proyectos. De acuerdo a lo anterior se identificaron los siguientes requerimientos para la gestión de proyectos de forma más ágil en la empresa Heinsohn Business Technology, se pueden observar en la **Tabla 3**.

Tabla 3. Requerimientos para la gestión de proyectos de forma ágil

Área de Conocimiento	Requerimiento
Alcance	Posibilidad de agregar, cambiar, o eliminar requisitos de manera flexible.
Comunicaciones	Retroalimentación temprana y constante por parte del cliente.
Tiempo y Costo	Indicación temprana de los problemas.
Tiempos	Desarrollo incremental en lugar de una única entrega completa al final del proyecto.
Alcance	Reducción de artefactos intermedios que no dan valor añadido al producto final.
Calidad	Reducción de desperdicios del producto y del proceso
Comunicaciones	Participación del cliente a lo largo del proceso de desarrollo con el fin de entregar un producto final acorde a las expectativas.
Riesgos	Tener entregables definidas en cada iteración que permitan medir de manera temprana del retorno sobre la inversión.
Tiempo y Costo	Reducir costo y tiempo invertido en reprocesos.
Calidad	Aumentar la eficiencia y demostrarles valor a los clientes e interesados.
Comunicaciones	Colaboración y valores compartidos.

1.4 Estudio de metodologías ágiles para la gestión de proyectos de desarrollo de software

El desarrollo ágil de software evolucionó desde la década de 1990 como reacción a los métodos de desarrollo en "cascada", caracterizados por ser muy estructurados y estrictos, que conducían a excesos en burocracia lo cual conducía a que los procesos fuesen lentos y costosos. Aunque los métodos de desarrollo iterativos ágiles pueden verse como un retroceso a las prácticas formales de desarrollo de software, hoy se ven conformes con las tendencias "lean" mundiales del cero desperdicio. [4]

El desarrollo de los métodos ágiles tuvo lugar hace muchos años y por múltiples autores. Como resultado de esto, se tienen inconsistencias en la terminología y

múltiples metodologías. A continuación, se establece un escenario para comprender que es la dirección de proyectos ágiles y explicar algunos de los conceptos clave que es necesario conocer, empezando con el manifiesto ágil. [4]

1.4.1 El Manifiesto Ágil

El manifiesto ágil surge como resultado de un encuentro en febrero de 2011 que reunió a un gran número de expertos en software y metodologías, los cuales definieron el manifiesto ágil y los principios ágiles. El manifiesto ágil propone un enfoque orientado a la participación de los usuarios y clientes, más que hacia los procesos y herramientas, trabajando más en el software y menos en la documentación, colaborando más con los clientes en vez de estar negociando y respondiendo a los cambios sacrificando el plan de trabajo si es necesario. [3]



Figura 2. Cuatro Valores del Manifiesto Ágil para el Desarrollo de Software
Fuente: Manifiesto Ágil, 2011.

La **Figura 2** muestra que, aunque existe valor en los elementos de la derecha, son más valorados los elementos de la izquierda.

1.4.1.1 Cuatro Valores del Manifiesto Ágil

Bajo la perspectiva del manifiesto ágil, se describen a continuación las cuatro sentencias.

1.4.1.1.1 Los individuos y las interacciones sobre los procesos y herramientas

Mientras que los procesos y las herramientas son necesarios en los proyectos, se debe intentar focalizar la atención del equipo en los individuos y las interacciones involucradas, ya que son las personas quienes llevan los proyectos y resuelven los problemas a través de procesos y herramientas, lo cual indica que hay que centrarse en desarrollar a los individuos involucrados en el proyecto, enfatizando en las interacciones productivas y efectivas, para preparar un proyecto para el éxito. Esto no quiere decir que los procesos y las herramientas no ayudan a completar exitosamente un proyecto, ellos son activos importantes, y siempre que haya un trasfondo de ingeniería tendida hacia la lógica y predictibilidad de los procesos. [4]

1.4.1.1.2 Software funcional sobre documentación exhaustiva

Este valor, habla de la necesidad de entrega de productos valiosos y de alta calidad en los proyectos de software, sin embargo muchas veces estos resultados quedan atrapados por la extensa documentación. Un software sin documentación, es realmente problemático y dificulta el soporte y el mantenimiento sin embargo la documentación exhaustiva sin software no tiene ningún valor para las organizaciones. Esta es la razón para que el manifiesto ágil oriente la atención en la utilidad del software por encima de la documentación exhaustiva como un recordatorio del porqué de estos proyectos – desarrollar algo útil para el cliente y el usuario final. [4]

1.4.1.1.3 Colaboración del cliente sobre negociación del contrato

Este valor enfatiza en ser flexibles y complacientes, en lugar de intransigentes y poco cooperativos. Se puede desarrollar el producto exactamente como se ha determinado en la especificación inicial, pero si el cliente cambia su idea inicial o sus prioridades, es mejor ser flexible y trabajar hacia la nueva meta, aunque sea opuesta a la originalmente establecida. Este desafío deriva de la naturaleza dinámica de los trabajadores con conocimientos especializados, especialmente en los sistemas de software; el software es intangible y difícil de referenciar, las compañías raramente crean los mismos sistemas dos veces, los negocios necesitan cambiar rápido, y la tecnología cambia rápidamente. En vez de afectar al cliente con un proceso de cambio de dirección, hay que reconocer desde el principio que las cosas van a cambiar, y debemos colaborar con el cliente durante el proyecto a través de una relación de confianza y modelos de contrato más flexibles. [4]

1.4.1.1.4 Respuesta a los cambios sobre seguir un plan inicial

Dado que los planes iniciales son a menudo inadecuados es preferible gastar más esfuerzo en tratar de responder adecuadamente a los cambios que se dan en el proyecto, en vez de reconducir el proyecto a la idea inicial, lo cual no quiere decir que

el manifiesto ágil plantea el abandonar la planificación y únicamente respondamos a los cambios, lo que se indica es que aunque hay que seguir un plan, hay que tener que el plan inicial fue realizado cuando no se conocía todo sobre el proyecto, y conforme el trabajo progresa, hay que actualizarlo. El responder a los cambios sobre seguir rígidamente el plan inicial, es especialmente significativo en los proyectos cuando es común un gran número de cambios. [4]

En resumen, el manifiesto ágil no es un conjunto de reglas a seguir, que nos digan que debemos hacer, es una guía para considerar los proyectos desde una perspectiva basada en el valor. [4]

1.4.1.2 Doce Principios Guía del Manifiesto Ágil [3]

Adicionalmente a los valores ágiles, los autores del Manifiesto ágil crearon doce principios para los métodos ágiles. Estos valores y principios son clave para entender los métodos ágiles ver **Figura 3**.



Figura 3. Los 12 Principios del Manifiesto Ágil para el Desarrollo de Software
Fuente: Manifiesto Ágil, 2011.

1.4.1.3 Métodos ágiles

Existen alrededor de una docena de métodos ágiles que se utilizan actualmente. Los más comunes son Scrum, Extreme Programming (XP), Feature Driven Development (FDD), Dynamic Systems Development Method (DSDM), y la familia de métodos Crystal (Crystal Clear, Crystal Yellow). Otros métodos están basados en los desarrollos Lean o Kanban.

1.5 Selección de una o más metodologías ágiles para guiar la identificación de los principales fundamentos para la gestión de proyectos de desarrollo de software

El objetivo de este apartado es seleccionar un total de **dos (2)** metodologías ágiles, que posteriormente serán caracterizadas y sobre las cuales se tomará referencia para la formulación de la metodología propuesta. Las metodologías que se han considerado para realizar la selección son: (Crystal Methods, Lean development, Scrum, Test Driven Development y Extreme Programming).

Para la selección de las metodologías a estudiar se tiene como premisa que es importante trabajar con metodologías lo suficientemente documentadas y que cuenten con algún tipo de certificación u entrenamiento, esto quiere decir que son también de alta aceptación en la industria del software.

Según estas condiciones se determinaron los siguientes criterios para realizar la valoración: **Metodología con mayor presencia en Internet, Metodología mejor documentada, Metodología con entrenamiento, Metodología con certificación.**

La valoración se llevó a cabo seleccionando para cada criterio las metodologías con mayor número de información y si contaban con algún tipo de entrenamiento o certificación, finalmente se contabilizaron los criterios para cada metodología y se determinaron las puntuaciones para seleccionar las más altas.

Tabla 4. Valoración de metodologías de acuerdo a criterios establecidos

Metodología	Artículos Científicos	Presencia en Bucadores	Libros en Español	Libros en Otros Idiomas	Entrenamiento	Certificación	Valoración
Crystal Methods	0	389.800	0	1	SI	NO	1
Lean development	2	39.290	0	0	SI	NO	1
Scrum	43	10.510.000	2	4	SI	SI	4
Test Driven Development	55	1.812.000	0	9	NO	NO	2
Extreme Programming	100	7.130.000	2	20	SI	NO	5

En vista de los resultados presentados en la **Tabla 4**, las metodologías seleccionadas fueron **Scrum y Extreme Programming.**

Para la caracterización de las metodologías se seguirán los puntos: Procesos, Roles y responsabilidades, Prácticas y herramientas, Entorno de uso.

A continuación se presentan en detalle las metodologías.

1.4.1.3.1 Scrum

Tabla 5. Metodología Scrum

PROCESO
<p>Se identificar tres fases en Scrum, planificación del sprint, seguimiento del Sprint y revisión del sprint.</p> <p>Planificación del Sprint</p> <ul style="list-style-type: none"> • Se define el Product Backlog, que consiste en una lista priorizada de requisitos del sistema, es un documento en constante actualización. • Planificación del sprint u iteración, es una jornada de trabajo previa al inicio de cualquier sprint y en la cual se determinan los objetivos y el trabajo que se debe cubrir. De esta reunión se obtiene una lista de tareas denominada Sprint Backlog. <p>Seguimiento del Sprint</p> <ul style="list-style-type: none"> • Se llevan a cabo breves reuniones diarias, para revisar el avance de las tareas y el trabajo que está previsto para la jornada. En estas reuniones solo están presentes el Scrum Master y el Equipo, se suelen realizar tres preguntas: <p>¿Qué trabajo se ha realizado desde la reunión anterior? ¿Trabajo que se va hacer hasta la próxima reunión? ¿Impedimentos que deben solventarse para proseguir con el trabajo?</p> <p>Revisión del Sprint</p> <ul style="list-style-type: none"> • Una vez finalizado el Sprint se realiza un análisis y revisión del incremento generado. Se presentan los resultados finales y se recomienda tener preparado un demo. <p>Existen múltiples razones para recomendar tener una demo al final de cada sprint, entre ellas destacamos la mejora del feedback con los interesados, reconocimiento del trabajo, un esfuerzo por finalizar las cosas o un correctivo en caso de identificar defectos.</p>
ROLES Y RESPONSABILIDADES
<p>Propietario del producto: Es el funcionario del proyecto conocedor del entorno de negocio del cliente y de la visión del producto. Es responsable de:</p> <ul style="list-style-type: none"> • Obtener el resultado de mayor valor para el cliente. • Toma de decisiones que afecten el resultado final, fecha de lanzamiento y retorno de inversión. • Proceso de adquisición del cliente. • Mantener el Product Backlog. <p>Scrum Master: Encargado de garantizar el funcionamiento de los procesos y de la metodología. Es el responsable de:</p> <ul style="list-style-type: none"> • Funcionamiento de modelo. • Interactuar tanto con el equipo como con el cliente y con los gestores.

Equipo de desarrollo: Es el equipo del proyecto y tiene la autoridad para decidir en las acciones necesarias y para auto-organizarse con la finalidad de alcanzar los objetivos del sprint. El equipo es responsable de:

- Estimar y ajustar el esfuerzo de las tareas del **Product Backlog**.
- Creación del **Sprint Backlog**.

Cliente: Participa en la creación del **Product Backlog**.

Gestor : Está a cargo de la toma de:

- Decisiones finales.
- Elección de objetivos y requisitos.
- Selección del propietario del producto.
- Negociación del **Product Backlog** junto con el **Scrum Master**.

PRÁCTICAS Y HERRAMIENTAS

Product Backlog: Se definen los requisitos del sistema o el trabajo a realizar a lo largo del proyecto. Está compuesto por una lista de requisitos de negocio y técnicos, actualizados y priorizados.

Sprint Backlog: Es una lista de trabajos que el equipo se compromete a realizar para generar el incremento previsto. Las tareas están asignadas a personas y tienen estimados el tiempo y los recursos necesarios.

Estimación de esfuerzo: Es un proceso iterativo en el cual las estimaciones de los ítems del **Product Backlog** son reajustadas acorde a la información obtenida en la última iteración.

Gráfico Burn-down: Es una herramienta para gestionar y seguir el trabajo de cada sprint y representa gráficamente el avance del sprint. En el gráfico el eje horizontal tiene una escala temporal que representa los días de duración del sprint y en el eje vertical otra escala que indica los puntos de historia o estimación de esfuerzo. De esta manera en las reuniones diarias el **Scrum Master** puede evaluar cuanto se ha avanzado y actualizar el gráfico.

Gráfico Burn-up: Herramienta de gestión y seguimiento que sirve al propietario del producto para controlar las versiones de producto previstas, las funcionalidades de cada una, la velocidad estimada, fechas probables de cada versión, margen de error previsto en las estimaciones y avance real. Herramienta de gestión y seguimiento para el propietario del producto.

Planning Poker o estimación de Poker: Juego que ayuda al equipo a establecer una estimación de las tareas en la reunión de inicio de sprint.

ENTORNO DE USO

Scrum es un método ideal para pequeños equipos, de diez o menos componentes. Se recomiendan equipos de cinco componentes, dividiendo los equipos en equipos más pequeños si fuera necesario.

Fuente. ScrumGuides.org, 2015. [5]

1.4.1.3.2 Extreme Programming (XP)

Tabla 6. Metodología Extreme Programming

PROCESO
<p>El ciclo de vida de XP está compuesto de seis fases:</p> <p>Fase de exploración: En esta fase los usuarios escriben las tarjetas de historia que ellos quieren que sean incluidas en la primera versión. Cada una de las tarjetas de historia describe una funcionalidad que será añadida al programa. El equipo de proyecto durante este tiempo se dedica a familiarizarse con las tecnologías y herramientas que utilizará a lo largo del proyecto, probando las herramientas y construyendo un prototipo simple para probar las posibilidades de la arquitectura.</p> <p>Fase de planificación: En esta fase se establece la prioridad de las diferentes historias y se acuerda el contenido de la primera entrega del proyecto. La estimación temporal se basa en un cálculo estimado por parte de los desarrolladores de cada una de las historias, la primera entrega no suele tardar más de dos meses en realizarse. La duración de esta fase no suele exceder el plazo de unos pocos días.</p> <p>Fase de iteraciones: Esta fase incluye la realización de diferentes fases antes de liberar la primera versión del producto. La planificación realizada en la etapa anterior se divide en diferentes iteraciones, de una duración variable entre una semana y cuatro. Los usuarios son los que deciden que historias se van a realizar en cada iteración, sabiendo que en la primera se suele realizar un sistema con la arquitectura de todo el sistema, seleccionando aquellas historias que ayuden a construirla. Las pruebas funcionales creadas por el cliente son ejecutadas al final de cada iteración, de tal manera que al final de esta fase obtenemos una versión lista para producción.</p> <p>Fase de producción: En esta fase se llevan a cabo un conjunto de pruebas extras, de rendimiento y funcionamiento que son necesarias antes de poder entregar el producto al cliente. Si se encuentran cambios importantes que se deban realizar al producto, se debe decidir si incluirlos en esta versión o dejarlos para posteriores. Las iteraciones de esta fase no deben durar más de tres semanas.</p> <p>Fase de mantenimiento: Una vez se libera la primera versión a los usuarios, el proyecto siempre se debe mantener en el entorno de producción. Esto supone un esfuerzo considerable en la fase de mantenimiento e incluso sea necesaria la contratación de nuevo personal para dar soporte a los clientes e incluso cambiar la estructura del equipo.</p> <p>Fase de cierre del proyecto: Es la fase en que los clientes ya no tienen más historias que deban ser implementadas. Es necesario evaluar que se satisfagan todas las necesidades de los clientes y otros aspectos como fiabilidad, rendimiento, etc. La documentación del proyecto se realiza en esta fase, ya que ni la arquitectura, ni el diseño, ni el código sufrirán cambio alguno.</p>
ROLES Y RESPONSABILIDADES
<p>Cliente: El cliente es el encargado de escribir las historias y las pruebas funcionales, y es el que decide cuando un requisito es satisfecho. También es el encargado de establecer la prioridad de las funcionalidades a implementar.</p> <p>Programador: Escribe tanto las pruebas como el código de la aplicación y se debe comunicar fluidamente y frecuentemente con sus compañeros.</p> <p>Probador: Ayuda a escribir las pruebas funcionales al cliente, ejecuta las pruebas funcionales, transmite los resultados de las pruebas y mantiene las herramientas.</p>

Rastreador: Es el encargado dar el “feedback”. Se encarga de seguir las estimaciones y de ir avisando de las desviaciones, para mejorar las futuras estimaciones. También se encarga de realizar el seguimiento de las iteraciones y valorar si los objetivos se pueden alcanzar con los recursos disponibles (tiempo, personal) o si es necesario hacer algún cambio.

Coach o tutor: Es la persona responsable del proceso Extreme Programming (XP). Es importante que tenga conocimientos y experiencia para guiar y ayudar al equipo a adaptarse a la metodología.

Consultor: Es un miembro externo que tiene los conocimientos técnicos necesarios. El consultor guía al equipo para solucionar los problemas específicos.

Gestor o Manager: Es el que toma las decisiones críticas y está en permanente contacto con el equipo para poder discernir las diferentes situaciones críticas.

PRÁCTICAS Y HERRAMIENTAS

Extreme Programming dispone de un gran abanico de técnicas y prácticas, muchas de ellas que han sido escogidas de diferentes metodologías existentes que tenían suficientemente probada su eficiencia.

El juego de la planificación: El equipo de desarrolladores estima el esfuerzo necesario para implementar las historias y los clientes determinan los objetivos y tiempos de entrega.

Historias de usuario: Son los requisitos del sistema formulados en una o dos sentencias, en el lenguaje común del cliente.

Cortas y pequeñas iteraciones: Un sistema simple se libera cada dos o tres meses y las diferentes versiones del mismo se suceden en periodos no superiores al mes.

Metáforas: El sistema se define utilizando un conjunto de metáforas acordadas entre el cliente y los programadores. Esta historia compartida guiará todo el proceso describiendo cómo funciona el sistema.

Diseño simple: Es de gran importancia a la obtención de diseños simples que se puedan implementar rápidamente, evitando diseños complejos y código extra.

Pruebas: Las pruebas unitarias se escriben antes que el código y están ejecutando continuamente. Las pruebas funcionales las escriben los clientes.

Refactorizar: Reestructurar el sistema eliminando duplicaciones, mejorando la comunicación, simplificando y añadiendo flexibilidad.

Programación por pares: Es la técnica que promulga que dos personas escriban código en el mismo ordenador.

Propiedad colectiva: Cualquiera puede compartir cualquier parte de código con cualquier otro componente del equipo.

Integración continua: Una nueva porción de código es integrada en el código fuente, tan pronto como este lista. El sistema es integrado y construido muchas veces a lo largo del día, todas las pruebas son ejecutadas y deben ser pasadas para aceptar la nueva porción de código.

40 horas a la semana: Se establece un máximo de 40 horas de trabajo semanales.

Disponibilidad del cliente: Los clientes deben estar disponibles y presentes cuando sean requeridos por el equipo de desarrollo.

Estándares de codificación: Reglas de codificación son establecidas y seguidas por los programadores. Se enfatiza la comunicación a través del código.

Espacios de trabajo abiertos: Un gran espacio, con cubículos es lo recomendable.

ENTORNO DE USO

XP es una metodología en la que es recomendable trabajar con equipos de tamaño medio o pequeños, es decir entre tres y veinte componentes como máximo. Se considera inapropiado que el equipo de trabajo se encuentre en ubicaciones distintas que no sean en la misma planta y oficina. También considera un factor determinante la cultura de la empresa, declarando que si observamos resistencia en cualquiera de las prácticas o técnicas de XP es muy probable que el proyecto fracase. Otro factor clave para utilizar con éxito XP es la tecnología, si esta no acepta cambios frecuentes o requiere de continuo feedback, es muy probable que no consigamos sacar el proyecto adelante.

Fuente. extremeprogramming.org, 2009. [6]

1.6 Identificación la complementariedad de la guía PMBOK y los estándares del PMI con las metodologías ágiles seleccionadas

A continuación se realiza un análisis de las metodologías ágiles investigadas dentro de los grupos de procesos y áreas de conocimiento de PMI con el fin de identificar la complementariedad.

1.6.1. GESTIÓN DE LA INTEGRACIÓN

En cuanto al inicio, no se tiene estipulada la definición de un acta. La planificación, se encuentra definida por la metodología dado los pasos a seguir. La ejecución se lleva a cabo de acuerdo a lo planificado y el monitoreo y control se realiza con las reuniones de seguimiento controlando los resultados estimados. Además se permite la inclusión de cambios en los requerimientos del cliente.

1.6.2. GESTIÓN DEL ALCANCE

La planificación se realiza por medio de un listado de requerimientos del producto donde se definen necesidades y funcionalidades del proyecto y en el plan de trabajo se especifican las tareas que el equipo debe realizar para generar el incremento previsto. Las tareas están asignadas a personas y tienen estimados el tiempo y los recursos necesarios. El monitoreo y control se realiza al final de cada iteración en una reunión que cumple la función de validación con el cliente. Las reuniones diarias con el equipo del proyecto permiten también dar seguimiento a las iteraciones y corregir problemas de ser necesario.

1.6.3. GESTIÓN DEL TIEMPO

La planificación se realiza al definir el plan de trabajo en el cual se establecen tareas, tiempos a alcanzar, recursos y se planean de acuerdo a la prioridad establecida con el

cliente. En las reuniones se asignan las responsabilidades de la entrega de cada tarea. Los recursos están definidos desde el inicio del proyecto. En las reuniones diarias se realiza control y seguimiento para determinar si el proyecto cumplirá con su línea base.

1.6.4. GESTIÓN DE LOS COSTOS

En las metodologías no se define nada al respecto, se deberían seguir utilizando las buenas prácticas planteadas por el PMI.

1.6.5. GESTIÓN DE LA CALIDAD

En la planificación del listado de requerimientos del producto se definen los criterios de calidad esperados. Durante la ejecución es responsabilidad del equipo del proyecto validar mediante pruebas que se cumpla con los criterios de calidad exigidos por el cliente. El monitoreo y control se lleva a cabo por medio de las etapas de definición y ejecución de pruebas en las cuales es posible detectar a tiempo defectos para la entrega de un producto de calidad.

1.6.6. GESTIÓN DE RECURSO HUMANO

En las metodologías no se define nada al respecto, se deberían seguir utilizando las buenas prácticas planteadas por el PMI.

1.6.7. GESTIÓN DE LAS COMUNICACIONES

En la fase de Inicio los involucrados son definidos con base en los roles establecidos en las metodologías y estos colaboran en la definición del listado de requerimientos del producto. El plan de comunicaciones se estructura de acuerdo a los tipos de reuniones planteados por el método. Durante la ejecución se utilizan reuniones cara a cara, telefónicas o vía web. Para el monitoreo y control se realizan informes de desempeño que son registrados para generar reportes de avance sobre las iteraciones.

1.6.8. GESTIÓN DE RIESGOS

En las metodologías no se define nada al respecto, se deberían seguir utilizando las buenas prácticas planteadas por el PMI. Pero de acuerdo al planteamiento que realizan ambas metodologías de trabajar con equipos de trabajo pequeños, de realizar entregas iterativas, mantener comunicaciones activas tanto con el equipo de trabajo como con el cliente es fácil monitorear y controlar los riesgos del proyecto.

1.6.9. GESTIÓN DE ADQUISICIONES

En las metodologías no se define nada al respecto, se deberían seguir utilizando las buenas prácticas planteadas por el PMI.

1. 7 Encontrar actividades del proceso para la gestión de proyectos de desarrollo de software basado en metodologías ágiles y enfoque PMI

A continuación se detallan las seis actividades propuestas para la metodología de gestión de proyectos de desarrollo de software, a través de principios y prácticas ágiles con enfoque PMI.

1.7.1. REALIZAR CONFIGURACIONES PREVIAS AL INICIO DEL PROYECTO

En la actividad realizar configuraciones previas al inicio del proyecto se identifica la participación del rol Gerente de Proyecto, se tiene como entradas la visión del proyecto y la estructura de desglose de trabajo (WBS) y como salidas el plan de gerencia de recursos, Plan de Trabajo y la Herramienta visual de Seguimiento. Está constituida por seis sub-tareas que se enumeran a continuación:

1.7.1.1. Seleccionar el equipo de trabajo: Se debe tener en cuenta que los participantes son multidisciplinarios, y participaran activamente en todas las etapas del ciclo de desarrollo. El tamaño del equipo está entre 8 y 16 personas.

1.7.1.2. Identificar iteraciones en paralelo: Para el caso de proyectos grandes, puede requerir trabajar con varias iteraciones en paralelo, para lo cual se debe asignar los equipos de trabajo acorde con su capacidad para conformar las diferentes iteraciones.

1.7.1.3. Establecer lugar de trabajo: Todos los miembros del equipo de trabajo deben estar en la misma localización física, para poder maximizar la comunicación. Si una parte del equipo se debe ubicar en una localización diferente se debe asegurar que se cuenta con los medios de comunicación adecuados para tener una interacción permanente y efectiva con el equipo.

1.7.1.4. Identificar dedicación de cada uno de los miembros del equipo: Se debe tener en cuenta la disponibilidad del equipo de trabajo, relacionado con: Horarios de trabajo, estudios, vacaciones, compromisos con otros proyectos. Trabajar en diferentes horarios, puede afectar la productividad del equipo. Asegurar en lo posible la estabilidad del Equipo durante la duración del proyecto.

1.7.1.5. Establecer duración de Iteración: La duración de una iteración se encuentra entre 2 y 4 semanas, la definición de tiempo está relacionada con la complejidad de los requerimientos que serán desarrollados durante el proyecto, y la capacidad para desagregarlos desde el punto de vista funcional.

1.7.1.6. Seleccionar Herramienta para el seguimiento y control de actividades: Es posible utilizar un tablero físico o virtual.

1.7.2. ELABORAR LISTADO DE REQUERIMIENTOS DEL PRODUCTO

La actividad elaborar listado de requerimiento del producto cuenta con la participación de los roles Cliente, Gerente de Proyecto y Equipo del Proyecto. Como entradas se identificaron Visión del Proyecto, Modelo de Proceso del Negocio e Información del cliente. Y como salidas se generan Casos de Uso, Requerimientos funcional, Revisiones de Pares y Defectos. Está constituida por seis sub-tareas que se enumeran a continuación:

1.7.2.1. Conocer el modelo de negocio y la visión: En esta etapa el equipo del proyecto (Ingenieros de Pruebas, Ingenieros de desarrollo, Arquitecto, Líder Técnico, entre otros) son informados del contexto del negocio (Alcance del proyecto, propósito, tecnología a utilizar, restricciones, documentación suministrada por el cliente). El “QUE” se desea implementar, enmarcado en unos requerimientos de producto a Alto Nivel.

1.7.2.2. Identificar los requerimientos del producto: Los requerimientos se deben expresa en términos funcionales del producto. Como buena práctica se recomienda el uso del prototipo donde se pueda explicar el requerimiento.

1.7.2.3 Revisión de requerimientos: El objetivo de esta actividad es revisar los artefactos de requerimientos generados para garantizar que cumplen con los criterios de aceptación y soportan el proceso para el cual se han especificado.

1.7.2.4. Priorizar Requerimientos: El listado de requerimientos es ordenado por valor, riesgo, prioridad y necesidad. Los elementos de primer orden determinan las actividades de desarrollo inmediatas.

1.7.2.5. Revisión del Alcance vs requerimientos: De acuerdo al alcance inicialmente establecido, se revisan los requerimientos identificados para realizar los ajustes que sean del caso, bien sea porque se consideraron requerimientos que no hacen parte del alcance o porque el alcance requiere ser ajustado.

1.7.2.6. Aprobación del listado de requerimientos: El cliente, aprueba los requerimientos y el orden asignado. Se obtiene la línea base. A medida que el producto es utilizado, se incrementa su valor y se proporciona retroalimentación, el listado de requerimientos puede ser más largo y exhaustivo. Los requerimientos nunca dejan de cambiar, así que listado de requerimientos es un artefacto vivo. Los cambios en los requerimientos de negocio, las condiciones del mercado, o la tecnología podrían causar cambios.

1.7.3. PLANEAR ITERACIÓN

La actividad planear proyecto cuenta con la intervención de los roles Gerente de Proyecto, Representante del Producto y Equipo del Proyecto. Como entradas se identificaron Visión del Proyecto, Estructura de desglose de Trabajo (WBS), Lista de Requerimientos y Plan de Recursos y como salidas Plan de Recursos, Plan de Trabajo Iteración, Matriz Estimación y Control de Riegos. Esta es una actividad enmarcada dentro del ciclo para la construcción del producto es por esta razón que para cada iteración es necesario realizar nuevamente o reevaluar las siguientes siete sub-tareas.

1.7.3.1. Asignación de Roles y Responsabilidades: En esta tarea, se asignan los roles requeridos. Por ejemplo el Gerente de Proyecto que es el facilitador del proceso, con conocimiento en el sistema de calidad y su aplicación en las diferentes áreas del proceso, su principal actividad es remover los impedimentos, ejerce actividades de gestión de proyectos. El Representante del Producto que representa la voz del cliente. Ayuda en la definición de los requerimientos y los prioriza. Debe tener la capacidad de tomar decisiones, para cambiar de prioridad los requerimientos y/o aceptar un incremento del producto. Al inicio de cada iteración se deberá reevaluar los requerimientos para asegurarse que en las próximas iteraciones los requerimientos con mayor prioridad son los que permitan dar un mayor valor. También participa en la presentación del resultado obtenido tras cada iteración validando que las soluciones implantadas cumplen con los criterios establecidos. Y se asignan la responsabilidades al equipo del proyecto a cada miembro del equipo de trabajo se le asignan sus tareas en las cuales tiene el derecho para hacer todo lo necesario (dentro de los límites de las guías del proyecto) para alcanzar el objetivo de la iteración, organizarse a sí mismo y a su trabajo, y debe finalmente realizar demostraciones del resultado del trabajo.

1.7.3.2. Seleccionar los requerimientos para la iteración: En esta tarea, el equipo de trabajo (Ingenieros de Pruebas, Ingenieros de desarrollo, Arquitecto, Líder Técnico, entre otros) Selecciona en conjunto los requerimientos que consideran pueden ser realizados en la iteración. La duración de la iteración ha sido previamente definida y se puede dar en semanas, días u horas.

1.7.3.3. Estimar requerimientos: Se puede utilizar la técnica de estimación que considere aplicable al Proyecto. En la cual para cada requerimiento se estima el esfuerzo que el equipo considera se requiere para realizar un requerimiento de forma completa.

1.7.3.4. Estimar la velocidad del equipo: Si es la primera iteración, como no se ha realizado ningún trabajo aún, se puede utilizar la técnica llamada "planificación basada en el compromiso". Teniendo en cuenta la duración de la iteración, se pregunta al equipo si se pueden comprometer con el requerimiento, hasta que el equipo ya no puede comprometerse más. Normalmente la velocidad del equipo se define por el Número de Puntos Funcionales / Requerimiento por Iteración.

1.7.3.5. Generar Iteración: Una vez estimado el esfuerzo se valida contra la duración estándar de la iteración, el tiempo no puede ser mayor. Si es mayor se debe ajustar eliminando los requerimientos de acuerdo al orden asignado hasta que el esfuerzo se

ajuste al tamaño de la iteración. Adicionalmente se debe actualiza el listado de requerimientos, definir cuál es el objetivo a cumplir cuando éste incremento de software se haya terminado y definir hora/sitio de realización de reuniones diarias preferiblemente al inicio de la jornada. Se recomienda la utilización de una herramienta como el **Grafico Burn-down** planteado por la metodología Scrum que ayuda al equipo en el monitoreo de su progreso y es el indicador sobre las posibilidades de terminar la iteración. El gráfico muestra cuantos puntos de Historia de Usuario o de Requerimiento faltan por hacer.

1.7.3.6. Identificar riesgos: Esta tarea se realiza para identificar y administrar de forma permanente los riesgos del proyecto.

1.7.3.7. Realizar la planeación de la iteración: Teniendo en cuenta la Estructura de desglose de Trabajo (WBS), la disponibilidad de recursos, la relación de actividades y tareas identificadas para la iteración, se realiza el cronograma de trabajo. Teniendo en cuenta los siguientes pasos: Desagregar requerimientos en actividades (tareas), Asignar responsables de cada actividad, Actualizar Herramienta / Tablero de (tarea), Definir Hitos / Fechas límite y Actualizar Herramienta / Tablero.

1.7.4. EJECUTAR ITERACIÓN

En la actividad ejecutar iteración participan los roles Gerente de Proyecto y Equipo de Proyecto, como entrada principal se tiene el Plan de trabajo de la Iteración y como salidas se presenta el avance en la Herramienta visual de seguimiento. En esta actividad se realiza el monitoreo y control continuo durante la construcción del producto y se ejecutan las siguientes cinco sub- tareas:

1.7.4.1. Realizar reunión diaria: Todos los días durante 15 minutos el equipo se reúne para revisar el avance de la iteración. La reunión diaria se debe realizar todos los días a la misma hora y en el mismo lugar, preferiblemente en frente del tablero de seguimiento, con todos los participantes de proyecto.

1.7.4.2. Reportar actividades: Cada integrante del equipo, debe indicar qué tareas realizó el día anterior, qué tareas va a realizar el día de hoy y qué problemas ha tenido. Los problemas pueden generar bloqueos en algunas tareas. En algunos casos cuando se reportan problemas los otros miembros del equipo tratarán de darle solución. No se debe permitir el análisis de problemas durante la reunión diaria; inmediatamente después de concluir la reunión diaria, se programan las reuniones necesarias para la revisión de problemas y bloqueos. La reunión diaria NO es una reunión de seguimiento, es una reunión para sincronizar el trabajo del equipo y detectar impedimentos, que puedan afectar el desarrollo de la iteración.

1.7.4.3. Actualizar tablero de seguimiento: A medida que cada miembro del equipo realiza el reporte de sus actividades, se actualiza el tablero de seguimiento bien sea físico o virtual.

1.7.4.4. Actualizar tiempo faltante iteración: De acuerdo al reporte se actualiza el tiempo faltante para terminar la iteración.

1.7.4.5. Resolver Impedimentos: Terminada la reunión el gerente del proyecto, atiende los impedimentos reportados durante la reunión diaria, planeando las actividades a realizar por el equipo de trabajo. En algunos casos requerirá incluirse como una tarea no planeada en el tablero y asignar responsables.

1.7.5. CERRAR ITERACIÓN

En la actividad cerrar iteración participan los roles Gerente de Proyecto, Equipo de Proyecto y Cliente como entrada de esta actividad se presenta el avance en la Herramienta visual de seguimiento y como salida los entregables a cliente de la iteración. En esta etapa se hace la presentación de los requerimientos que fueron implementados y los que no. De la misma manera se presentan los posibles controles de cambio o nuevos requerimientos que deben ser incluidos. Se deben ejecutar las siguientes ocho sub-tareas:

1.7.5.1. Revisar y relacionar requerimientos terminados: Se debe verificar, antes de realizar la presentación de la iteración, si los requerimientos que han sido marcados como terminados, cumplen con los criterios acordados por en la planeación. Al inicio de presentación se indican cuáles de los requerimientos que hacían parte de la iteración fueron terminados, cuáles no planeados se adicionaron, y los requerimientos que no fueron terminados.

1.7.5.2. Presentar demo del producto: Se hace una presentación en el ambiente asignado para tal fin del incremento de Software terminado.

1.7.5.3. Documentar nuevos requerimientos: Durante la presentación del producto se pueden generar nuevos requerimientos que no fueron contemplados en el listado de requerimientos inicial, dichos requerimientos deben ser documentados, priorizados e incorporados.

1.7.5.4. Registrar defectos: Durante la presentación del producto se pueden presentar defectos relacionados con el comportamiento esperado del software. El equipo de trabajo puede decidir si lo trabaja como un defecto o como un nuevo requerimiento que debe ser incluido.

1.7.5.5. Aprobar / Rechazar iteración: Dependiendo de los resultados el Representante del Producto acepta el incremento de Software. Puede aceptarlo con observaciones, las cuales deben ser consideradas y analizadas para la próxima iteración. O puede rechazar el incremento dado los resultados de la presentación.

1.7.5.6. Actualizar tablero de seguimiento: Se realiza la actualización de los planes, cerrando la iteración y limpiando el tablero de seguimiento.

1.7.5.7. Anunciar próxima iteración: Se define la fecha en la cual se va a iniciar la próxima iteración.

1.7.8.8. Realizar retrospectiva de la iteración: En esta etapa se hace una evaluación de cómo se llevó a cabo el proceso, y se plantean acciones de mejora.

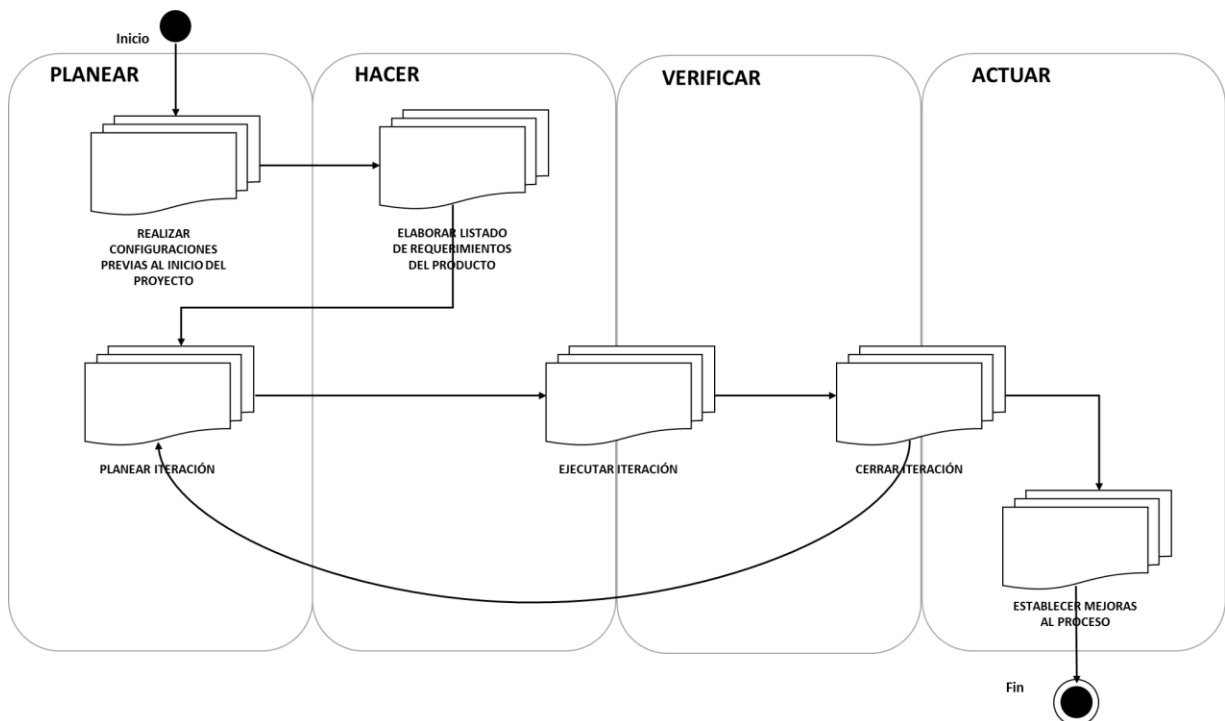
1.7.6. ESTABLECER MEJORAS AL PROCESO

La actividad establecer mejora al proceso puede ser ejecutada por cualquier funcionario de la organización que haga uso de la metodología y requiera realizar alguna sugerencia o petición, como salida de esta actividad se genera el registro de una acción correctiva o de mejora que en caso de ser aprobada por el área de calidad genera la optimización de la metodología.

2. RESULTADOS Y DISCUSIONES

2.1. Formulación de metodología para la gestión de proyectos de desarrollo de software, a través de principios y prácticas ágiles con enfoque PMI

Se formula la metodología para la gestión de proyectos de desarrollo de software a través de principios y prácticas ágiles con enfoque PMI enmarcada dentro de un ciclo PHVA (Planear, Hacer, Verificar y Actuar), donde la primera actividad es realizar las configuraciones previas al inicio del proyecto (*Ver numeral 1.7.1*) en la cual se establecen los lineamientos, equipo de trabajo y herramientas que se van a utilizar a lo largo del mismo, seguida de la actividad elaborar listado de requerimientos del producto (*Ver numeral 1.7.2*) donde se define en conjunto con el cliente los requerimientos, prioridad y plan de trabajo. Luego se observan tres actividades cíclicas planear, ejecutar y cerrar iteración (*Ver numerales 1.7.3, 1.7.4 y 1.7.5*) que encierran la construcción del producto final de manera incremental y con entregas tangibles a cliente en el cierre de cada iteración y finalmente un actividad que promueve la mejora y optimización del proceso (*Ver numeral 1.7.6*).



3. CONCLUSIONES

- PMI presenta un manejo integral del proyecto y no se centra solo en el ciclo del producto es por esta razón que las metodologías ágiles presentan un esquema más detallado de las fases necesarias para la construcción del producto software.
- La mayor fortaleza de las metodologías ágiles investigadas es que permiten al cliente ver productos terminados rápido sin esperar por un producto final completo.
- PMI no es una receta para seguir al pie de la letra para que los proyectos sean exitosos, esto permite adecuarlo a cualquier metodología propia y tomar como ejemplo las buenas prácticas.
- Las metodologías ágiles enmarcadas en esta investigación están orientadas al desarrollo de software rápido, en cambio PMI no se especializa en un solo tipo de proyecto.
- Las metodologías ágiles se centran en generar un producto rápido sacrificando muchos factores que pueden afectar el proyecto, dejando muchos vacíos que deben ser complementados con otras metodologías y buenas prácticas.
- Para el éxito de un proyecto gestionado mediante metodologías ágiles es necesario contar con el compromiso y participación continua del cliente.

REFERENCIAS BIBLIOGRAFICAS

- [1]. Project Management Institute PMI. (s.f.). Preguntas frecuentes del PMI-ACP®. *Project Management Institute PMI*, <https://americalatina.pmi.org/latam/CertificationsAndCredentials/PMI-ACP/PMI-ACPEXamPreparation/AgileFAQ.aspx>
- [2]. HUP, Sistema de gestión de calidad Heinsohn Business Technology.
- [3]. Authors: The Agile Manifesto. (2001). Manifiesto por el Desarrollo Ágil de Software. <http://www.agilemanifesto.org/iso/es/>.
- [4]. Universidad de Santander. Metodos Ágiles. http://tic.uis.edu.co/ava/pluginfile.php/182583/mod_resource/content/1/Metodos%20agiles.pdf
- [5]. Ken Schwaber, J. S. (2015 ScrumGuides.org.). SCRUM GUIDES. <http://www.scrumguides.org/>.
- [6]. Wells, D. (2009). Extreme Programming. <http://www.extremeprogramming.org/>.