

Integración de la Guía PMBOK con los Métodos Ágiles de Desarrollo de Software

Francisco Azofeifa Porras, Benji Marín Quesada

Escuela de Ingeniería,
Universidad Latinoamericana de Ciencia y Tecnología,
ULACIT, Urbanización Tournón, 10235-1000
San José, Costa Rica
[fazofeifap452,bmarinq711]@ulacit.ac.cr
<http://www.ulacit.ac.cr>

Abstract. Cada vez son más las empresas e individuos que utilizan métodos tradicionales y no tan tradicionales para gestionar proyectos de desarrollo de software. Ambos enfoques brindan beneficios y tienen a su vez puntos bajos, lo que incide en su elección. Este artículo hace mención al crecimiento de ambas metodologías de gestión de proyectos y los problemas que se enfrentan durante su ejecución.

El propósito de la siguiente investigación es presentar una guía de aplicación, que permita la integración de las prácticas sugeridas en la guía PMBok® específicamente en los cinco procesos de gestión de proyectos, a la gestión de proyectos de desarrollo de software que utilicen Metodologías Ágiles, tales como SCRUM, Xtreme Programming o Kanban.

Keywords: Agile, PMBOK, Project Management Institute, Gestión de Proyectos, SCRUM, eXtreme Programming, Kanban

1 Introducción

Hoy en día las empresas se encuentran en un mundo globalizado donde la competencia es cada vez más encarnizada. Es obligación de cada organización mejorar sus técnicas de gestión en todo lo que realizan, si desean mantenerse vigentes. La gestión de proyectos es seguramente el mayor pilar que las empresas, sean estas pequeñas ideas de negocio, o bien, empresas grandes y bien posicionadas, deben gestionar con el mayor cuidado la ejecución de estos.

Con el propósito de definir un adecuado fundamento teórico que sirva como estructura para el presente artículo, se hizo la revisión de múltiples fuentes bibliográficas relacionadas con el desarrollo de software que siguen, en mayor o en menor grado, los conceptos propuestos en el Manifiesto para el Desarrollo Ágil de Software. Así mismo, se utilizó la Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK) Quinta Edición, por ser la última versión disponible.

2 Estado del Arte

Día a día las empresas y personas se dedican a generar Valor mediante creación de diversos productos o servicios tales como aplicativos de software. Como describe el PMBok (Project Management Institute, 2013), dichas creaciones se pueden enmarcar con el nombre de proyecto, ya que éstos son definidos como “. . . un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único” (Project Management Institute, 2013). Asimismo, su propia naturaleza implica que los proyectos tienen un principio y un final definidos. Lastimosamente, no siempre ese final es el esperado y muchos proyectos tienden al fracaso.

Una causa común de problemas en el desarrollo de software es que el producto final sea precisamente, “*lo que el cliente pidió*”. En las metodologías más comunes de desarrollo de software, el usuario final, llámesele Cliente, brinda una serie de ideas y solicitudes. A su vez, el analista toma esas ideas, por lo general abstractas, y entre ambos definen una serie de requerimientos formales con los que se espera dar un fin apropiado a ese proyecto que el cliente requiere. Sin embargo, al final del camino, muchas veces el proyecto no es lo que imaginó originalmente el cliente y los resultados pueden ser negativos para todas las partes involucradas.

En la actualidad, un alto porcentaje de proyectos fallan o presentan problemas durante su ejecución. De acuerdo al estudio citado por Jim Johnson, fundador del Grupo Standish, los resultados de la encuesta CHAOS del año 2004 (The Standish Group, 2005) indicaron que la mayoría de los proyectos de desarrollo de software presentaron grandes problemas, o bien, fallaron. Dicha encuesta mostró que un 53% de los proyectos no fueron lo que sus clientes desearon, 18% fallaron y tan solo un 29% logra su cometido.

Todos los proyectos, independientemente de su naturaleza, se ven delimitados por tres grandes fuerzas: El Presupuesto, que determina la disponibilidad de recursos económicos, y como tal, materiales y humanos; El Tiempo, el cual, como se citó anteriormente convierte el proyecto en un esfuerzo temporal; y El Alcance, que nos dice el Qué vamos a hacer. La conjunción de estos tres factores determinan un cuarto elemento: La Calidad. Todos desean un producto o servicio que se obtenga en el menor tiempo, que sea barato, que sea bueno y por supuesto, que cumpla con las expectativas de funcionalidad.

Tradicionalmente, el desarrollo de software se ha realizado mediante el modelo de “Cascada” (Bassil, 2012), en el cual todas las actividades se realizan de manera secuencial. El orden típico de dichas tareas sería: Análisis, Diseño, Implementación, Pruebas y Mantenimiento. Este modelo es funcional y ha estado vigente muchísimos años, sin embargo genera el problema que se ha venido discutiendo: el cliente tiene contacto con el producto técnicamente terminado, cuando las modificaciones representan un enorme esfuerzo en refactorizar miles de líneas de código. Esta distancia priva a ambos, tanto al grupo de desarrollo

como al cliente, de toda la retroalimentación que ambos tienen para dar en caso de trabajar más en colaboración.

En este contexto, surge una necesidad superior de utilizar metodologías de desarrollo flexibles, que permitan acortar los tiempos de desarrollo sin minar la calidad, que siendo dinámicas admitan incorporaciones y mejoras a la idea original durante el trayecto por parte del cliente y que, finalmente, brinden un producto lo más cercano a las necesidades reales del cliente. Así aparecen las denominadas metodologías de desarrollo ligero que posteriormente serían agrupadas y denominadas como Metodologías Ágiles, a las cuales nos referiremos simplemente como Agile ¹ para simplificar.

Ya desde inicios de los años 90, varias personas comenzaron a crear metodologías o marcos de trabajo que permitieran gestionar proyectos de desarrollo de software que fuera más adaptativos a las necesidades reales de los clientes. Para el año 2001 un grupo de personas se reúnen para redactar los que se conoce como el Manifiesto para el Desarrollo Ágil de Software (Beck et al., 2001), o simplemente Manifiesto Ágil. A partir de aquí, los desarrolladores han comenzado a realizar cambios significativos a los proyectos a partir de diferentes marcos de trabajo.

El Manifiesto Ágil no reniega de las bondades de los métodos tradicionales. Por el contrario, valora aspectos relacionados con esos métodos (situados en la siguiente lista a la derecha), pero le da un valor aún mayor a aquellos situados a la izquierda:

- Los individuos y las interacciones, sobre los procesos y herramientas.
- Software funcional, sobre la documentación completa.
- La colaboración con los clientes, sobre la negociación del contrato.
- Respuesta ante los cambios, sobre seguir un plan.

Dentro de los marcos de trabajo de desarrollo ágil de software, encontramos que métodos como Scrum, XP y Kanban están ganando popularidad en la gestión de proyectos de desarrollo de software. Día a día, cada vez más empresas están reportando mejoras de productividad y mayor satisfacción por parte de clientes con los productos de software desarrollados utilizando alguna de estas metodologías las cuales, por supuesto, no son excluyentes.

La encuesta “Agile Methodologies Survey” realizada en el año 2003 por Shine Technologies (Griffiths, 2004), sobre métodos Ágiles, la cual incluyó 131 organizaciones del ramo de desarrollo de software, arrojó datos interesantes:

- Un 93% dijo que el Trabajo en Equipo mejoró

¹ AGILE: Desarrollo de Software Ágil, es un enfoque de gestión de proyectos de software donde los requerimientos y soluciones pueden cambiar conforme avanza el proyecto.

- El 88% percibió que la calidad del software desarrollado fue más alta.
- 83% experimentó una mayor satisfacción con el producto.

Los datos expuestos por la encuesta citada no son del todo extraños en virtud de los beneficios que brinda los marcos de trabajo ágiles a las organizaciones que los implementan. Dichos beneficios varían de empresa a empresa, de acuerdo al mayor o menor grado de integración de sus actividades diarias con dichos marcos de trabajo. Sin embargo, tal como indica Mark Layton (Layton, 2012) los siguientes son la mayoría de los beneficios que brindan las metodologías ágiles:

2.1 Mayor Calidad

Agile proporciona el marco para la retroalimentación continua entre el Dueño del Producto y el equipo de trabajo, por lo que la exposición del producto es la más alta posible durante la etapa de desarrollo, por lo que puede asegurarse la calidad. Agile ayuda a asegurar la calidad mediante las siguientes prácticas:

- Los requisitos del producto se elaboran y definen tan pronto es posible, y son del conocimiento de todo el equipo de trabajo.
- La incorporación de las pruebas diarias y la retroalimentación por parte del dueño del producto en el proceso de desarrollo permite al equipo de trabajo hacer frente a los problemas y buscar soluciones más satisfactorias para el dueño del producto.
- Mejora regular y continua del producto a través de las revisiones de sprint con las partes interesadas.
- Completando el trabajo mediante la definición de lo “hecho”, en otras palabras definiendo lo desarrollado, lo probado, lo integrado y lo documentado.

2.2 Disminución del tiempo de comercialización

Si comparamos la velocidad de entrega del producto entre Agile y las metodologías tradicionales desarrollo de software, vemos que disminuye el tiempo de entrega entre un 30% y 40%. Esta disminución puede atribuirse a la siguiente lista elementos:

- Habitualmente, en el modelo cascada las fases iniciales toman mucho tiempo. En Agile se agrega valor al incorporar un dueño del producto al equipo de trabajo, el cual brinda, no sólo especificaciones tempranas, sino además aclaraciones en tiempo real al equipo de trabajo.
- Se realiza una priorización dentro de los elementos a desarrollar, aquellos con un mayor grado de criticidad son desarrollados primero y pueden ver la luz al mercado mucho antes.
- El equipo de trabajo de Agile brinda al final de cada sprint o iteración, un producto completo, probado y con calidad que puede ser utilizado en el mercado.

2.3 El aumento de rendimiento de la inversión

La disminución en el tiempo de comercialización, citada anteriormente, es una de los puntos clave para que los proyectos Agile brinden un mayor retorno de la inversión (ROI). Debido a que los ingresos y otros beneficios empiezan a llegar más pronto significa mayor rendimiento total en el tiempo. Este es un principio básico al calcular el valor actual neto (VAN). Adicionalmente a los beneficios producidos por la disminución del tiempo de salida al mercado, el retorno de la inversión también aumenta por:

- Retroalimentación regular a través de revisiones de sprint directamente con los clientes, lo permite correcciones tempranas en caso de que se requiera, y a su vez, esto implica una reducción de costos.
- La revisión constante implica menos trabajo perdido y entregas más rápidas.
- Un aspecto muy importante es que si un proyecto Agile va a fallar, se puede detectar ese fracaso de manera mucho más temprana que en un proyecto en Cascada.

2.4 Mayor satisfacción del cliente

Equipos de Agile se comprometen a la producción de productos y servicios que satisfagan a los clientes. Agile permite mayor satisfacción por parte de clientes y patrocinadores de proyecto a través de los siguientes aspectos:

- La colaboración con los clientes y patrocinadores, manteniéndolos involucrados y comprometidos durante la ejecución del proyecto.
- Tener un dueño del producto que es un experto en los requisitos del producto y las necesidades del cliente junto al equipo de desarrollo.
- Mantener la cartera de solicitudes de productos actualizados, estableciendo prioridades para responder rápidamente a los cambios.
- Demostrando la funcionalidad del trabajo realizado a patrocinadores y clientes en cada revisión del sprint.
- La entrega de productos a clientes finales más rápido y más a menudo con cada lanzamiento en lugar de todos a la vez en el final.
- Mediante la entrega de productos más rápido y más frecuentemente entre lanzamientos en lugar de una sola entrega final del proyecto.
- El cliente financia el proyecto de manera incremental, respondiendo a entregas periódicas, en lugar de una gran inversión inicial.

2.5 Equipo de trabajo más motivado

Agile se centra en los equipos auto gestionados. Esto permite que cada miembro del equipo de trabajo sea parte la toma decisiones y genere un mayor compromiso hacia los objetivos del proyecto y de la organización. Agile mejora la motivación de los miembros del equipo de la siguiente manera:

- Ser parte de un equipo de auto-gestionado y auto-organizado incentiva la creatividad.
- Los equipos de desarrollo pueden organizar su estructura de equipo en torno a las personas con estilos y personalidades de trabajo diferentes.
- Los equipos de trabajo de Agile pueden tomar decisiones adaptadas para proporcionar equilibrio entre vida profesional y personal los miembros del equipo.
- Tener una relación entre iguales con el dueño del producto alinea las prioridades técnicas y de negocios y rompe las barreras organizacionales.
- Tener un Agile master (Leffingwell, 2011), que sirve al equipo de Agile, elimina los obstáculos y protege el equipo de desarrollo de interferencias externas.
- Mantener conversaciones cara a cara ayuda a reducir la frustración de la falta de comunicación.

2.6 Mejora la visibilidad de los avances y la exposición

En proyectos gestionados mediante Agile, todos los miembros del equipo del proyecto, tanto equipo desarrollador como patrocinadores y clientes pueden conocer exactamente el grado de avance del proyecto en un momento dado, lo que permite determinar tempranamente cualquier problema y cómo cambiar el rumbo:

- Existe una gran transparencia mediante una comunicación abierta y honesta entre el equipo de trabajo de Agile, los interesados, los clientes, y cualquier otra persona dentro de una organización que quiere saber sobre un proyecto.
- Las reuniones diarias permiten conocer todos los días los avances y obstáculos enfrentados.
- Las reuniones diarias permiten al equipo de desarrollo organizar e identificar las tareas de mayor prioridad para el día.
- Las revisiones de sprint permiten a los miembros del equipo de Agile para identificar lo que está funcionando bien y qué no, y a su vez generar planes de mejora.
- Demostrar logros en los exámenes de sprint. Cualquier persona dentro de una organización

2.7 Mayor control de proyectos

Equipos de Agile tienen múltiples opciones para controlar el desempeño del proyecto y hacer las correcciones necesarias:

- Ajuste de prioridades lo largo del proyecto en cada sprint, en lugar de gestionar dichas prioridades solo en hitos, permite al equipo de trabajo responder mejor a situaciones imprevistas.
- Considerando el cambio como un elemento natural, permite al equipo del proyecto reaccionar a factores externos de manera más oportuna y natural.

- Las reuniones diarias permiten al equipo de Agile abordar rápidamente los problemas que puedan surgir, y generar respuestas rápidas a dichos problemas.
- La comunicación cara a cara mejor la comunicación y elimina otras barreras.
- Las revisiones del sprint permite al dueño del producto revisar periódicamente el avance del proyecto y asegurarse que todo va por buen camino.
- Las reuniones diarias y las revisiones de sprint permiten a todos los involucrados - el equipo de desarrollo, dueño del producto, Agile Master, y otros grupos de interés - ejercer el control y, por ende, crear mejores productos.

2.8 Reducción del riesgo

Agile ayuda a mitigar el riesgo de fracaso del proyecto a mediante la entrega de producto tangible, de manera temprana y brindando la posibilidad de visualizar problemas al equipo de trabajo de Agile:

- Contar con los elementos de mayor riesgo realizado primero proporciona la pista más larga de trabajar a través de cuestiones o falla temprana y económica.
- Desarrollar en los sprints, asegurando un corto período de tiempo entre la inversión inicial del proyecto y, o bien no haber rápido o la validación de que un producto o un enfoque funcionarán.
- Tener un mínimo de la subasta de productos de trabajo a partir de la primera carrera, por lo que incluso si un proyecto se termina, los requisitos de mayor valor y de riesgo han sido desarrollados y podrían ser entregado al cliente si lo desea.
- El desarrollo de los requisitos de la definición de hecho en cada sprint para que los patrocinadores del proyecto han completado, características útiles, independientemente de lo que pueda ocurrir con el proyecto en el futuro.
- Proporcionar retroalimentación constante sobre los productos y procesos.

Estos elementos brindan una justificación más que válida para el uso de metodologías Ágiles en cualquier proyecto de software. Sin embargo, el uso de SCRUM, XP o Kanban como metodologías de desarrollo también presentan algunos puntos bajos (Sharma, Sarkar, & Gupta, 2012) que es importante citar: Gasto de tiempo en refactorización del código producto de nuevas solicitudes del dueño del producto; además tiende a ser más útil para el Agile master que para el programador. Se requiere una alta participación del dueño del producto, el cual debe no solo contar con el tiempo, sino además ser experto en la dinámica de negocio que genera la necesidad del producto. Caso contrario es muy probable que el proyecto se descarrile. Así mismo, al priorizarse el producto por encima de la documentación, esta puede llegar a ser escasa o nula.

Debido al aumento el uso de los métodos ágiles y un aumento similar en la adopción de enfoques tradicionales de gestión de proyectos, un número creciente de organizaciones están enfocadas en intentar conciliar ambos marcos de trabajo,

los cuales tienden a parecer diametralmente opuestos en puntos fundamentales.

Los métodos ágiles no son sólo una forma de desarrollar software, que a su vez puede gestionarse con métodos convencionales de administración de proyectos. Los métodos ágiles requieren la adopción de determinados enfoques de gestión de proyectos para que se puedan llevar a cabo. El ámbito de aplicación de los métodos ágiles va más allá de las actividades del equipo de desarrollo y cambia la forma en que los patrocinadores, usuarios y otras partes interesadas participan de los proyectos, independientemente de su naturaleza.

Los enfoques de desarrollo ágiles también emplean procesos de planificación, de ejecución y, de manera similar, de control, que son fundamentalmente diferentes y que lógicamente son esenciales para alcanzar el éxito. Conocer las diferencias entre los dos modelos de gestión de proyectos, AGILE y PMI, es fundamental.

Los métodos tradicionales de gestión de proyectos también están en aumento. Estadísticas de miembros del PMI (Project Management Institute, 2014) indican un aumento continuo de la popularidad; el crecimiento es superior a 20% al año, y aproximadamente un 25% entre 2012 y 2013 tan solo en Latinoamérica, y nada hace indicar que dicha tendencia baje. Este auge generó que el PMI desarrollara su guía de buenas prácticas con el propósito de brindar a sus miembros, denominado PMBOK. En su 5 revisión, el PMBOK incluye cinco procesos y nueve áreas de conocimiento. El presente artículo se enfocará en los cinco procesos que son:

- **Inicialización:** Es el inicio formal del proyecto, o bien de una etapa del mismo.
- **Planificación:** La definición del alcance, los objetivos y las métricas del proyecto del proyecto se realizan en este proceso.
- **Ejecución:** Procesos para completar el trabajo documentado en el Plan de Gestión de Proyectos.
- **Monitoreo y Control:** Esta etapa consta de todas las actividades de seguimiento y revisión de las diferentes etapas del proyecto.
- **Cierre:** Básicamente, es la formalización de cómo termina la fase actual del proyecto o el proyecto en sí mismo.

Otro resultado del aumento en la adopción de ambos marcos de trabajo o frameworks, tanto métodos ágiles y tradicionales, es la confusión que a menudo surge debido a la falta de coincidencia entre los principios y técnicas de cada uno de estos enfoques de gestión de proyectos.

3 Implementación de Procesos de PMBOK y Agile

Otro aspecto importante de notar respecto a la metodología del PMI ha sido su evolución. A través del tiempo los miembros del PMI han ido realizando mejoras

al PMBOK de manera tal que sirva en un mayor rango de proyectos. Parte de estas mejoras tiene que ver directamente con los ciclos de vida de los proyectos.

Ciclos de Vida Iterativos e Incrementales. Los ciclos de vida iterativos e incrementales son aquellos proyectos en los cuales, algunas de las fases del proyecto se repiten de manera intencionada con el propósito de adicionar o mejorar alguna característica. Dichas iteraciones desarrollan el bien o servicios por medio de una serie de ciclos que se repiten, cada repetición o iteración genera un incremento que se va añadiendo funcionalidad a ese producto.

Por su parte, los Ciclos de Vida Adaptativos (Project Management Institute, 2013) pretenden responder a exigencias de cambio durante la ejecución y fomentan la participación continua de interesados. Los ciclos de vida adaptativos tienen afinidad con los ciclos de vida iterativos e incrementales de los que se versó en el párrafo anterior, sin embargo, su principal diferencia radica en que las iteraciones son bastante más rápidas, con duraciones que varían entre de 2 y 4 semanas. Los proyectos con ciclos de vida adaptativos generalmente ejecutan varios procesos en cada iteración, aunque las iteraciones iniciales pueden centrarse más en las actividades de planificación.

El alcance del proyecto será determinado por un conjunto de requisitos y labores a realizar. Al inicio de cada iteración, el equipo de trabajo elige cuales de los requisitos la lista de requisitos se pueden entregar dentro de esa ventana de tiempo o iteración. Al final de cada una de esas iteraciones se debe tener un producto listo para su revisión por parte del cliente. Básicamente se espera no un producto perfecto, sino, un producto funcional. Este tipo de ciclo de vida implica una continua participación por parte de los representantes del patrocinador o cliente, ya que estos deben brindar retroalimentación continua durante la etapa de planificación y desarrollo, revisando adicionalmente todos los productos de las iteraciones, con el propósito de mantener el desarrollo del producto siempre lo más cercano a las necesidades de estos.

Generalmente se opta por los proyectos con ciclos de vida adaptativos en entornos que cambian rápidamente, cuando el alcance y los requisitos son complicados de definir desde una fase preliminar y cuando es esperable que se pueda generar pequeñas mejoras graduales al producto y generar mayor satisfacción en los interesados. Como se observa, el PMBOK se ha ajustado a las necesidades de adaptación de proyectos que pueden ser impredecibles y cambiantes. Dicha mejora calza de buena manera a las necesidades de un proyecto de desarrollo de software alineado a las metodologías ágiles.

Es importante pues, que se tenga claro que podemos obtener grandes beneficios si logramos una adecuada mezcla de los beneficios que se obtiene tanto de las Metodologías de Desarrollo Ágil como PMBOK. Ambos marcos de trabajo ofrecen una guía abierta de cómo gestionar un proyecto. Los métodos ágiles

brindan un marco de trabajo flexible y orientado a la productividad por encima de la formalidad. Sin embargo, ya se acotó la mejor que ha sufrido el mismo PMBOK, adicionando la posibilidad de gestionar ciclos de vida adaptativos, los cuales bien conducidos, puede brindar el valor agregado de contar con mejor documentación y controles a los métodos ágiles.

Para tal fin, se procede a realizar un mapeo de los procesos más importantes del PMBOK y relacionarlos con su equivalente en las metodologías ágiles.

3.1 Proceso de Iniciación

El proceso de inicio indicado en el PMBOK busca definir el presupuesto, el alcance (generalmente fijo) y un cronograma de actividades que abarque todo lo largo y ancho del proyecto. Esta información queda generalmente registrada en Carta de Constitución del Proyecto (Project Charter en inglés). En la misma carta de constitución debe quedar claro el mecanismo por medio del cual se entregará el producto. De manera homologa, el proceso de Iniciación de los Métodos Ágiles establece una visión común, así como las metas del equipo y se define una lista de actividades que deberá abarcar no más de 2 a 4 semanas.

Se deberá realizar una Carta de Constitución del Proyecto por única vez antes de la primera iteración o sprint, indicando la mayoría de los componentes de esta, y estableciendo cuál metodología se va a aplicar. Al utilizar los métodos ágiles, los conceptos de desarrollo con ciclos de vida iterativo y el desarrollo por incrementos regulares de software debe ser explicado claramente, tanto al Dueño del Producto como a los demás interesados, siguiendo el espíritu del Manifiesto Ágil.

3.2 Proceso de Planeación

En la fase de Planeación según la define el PMI, se ejecuta previo al desarrollo de cualquier actividad de la fase de Desarrollo. Aquí se planifica todo lo relacionado con la gestión de recursos humanos, gestión de riesgos, etc. Mientras tanto, los métodos ágiles se apoyan en la adaptabilidad para gestionar estos elementos de manera más continua, como mínimo, al inicio de cada iteración. El alto grado de valor que se le brinda a la comunicación cara a cara en los métodos ágiles limpia el camino para que la planificación de actividades, tanto en sprint como de manera diaria se dé de una manera más natural. Adicionalmente, la participación continua del Dueño del Producto conocer cómo cambian las prioridades de los interesados.

Siendo que la planificación es una fase crítica tanto para la gestión tradicional de proyectos, como para aquellos desarrollados con metodologías ágiles, no debe tomarse a la ligera. Sin embargo, enfocar toda la fuerza en esta fase es riesgoso, ya que es el momento en el que solo se tiene una idea general del problema a

resolver. Por el contrario, la planificación se deberá gestar de manera iterativa y continua durante todo el proyecto, aprovechando la retroalimentación que brinda el Dueño del Producto y los Interesados, la experiencia que acumulará el equipo de trabajo a lo largo de los Sprint.

Como tal no se debería contar con un Plan de Gestión del Proyecto de manera tan formal como se solicita en el PMBOK, donde se desarrolla como una fuente primaria de información para determinar el camino por donde se ejecutará, supervisará, controlará y se le dará cierre al proyecto. Por el contrario, la recopilación de requisitos si se vincula de manera directa entre el PMBOK y las Metodologías Ágiles por medio de las historias de usuarios. Gracias las Historias de Usuarios, las cuales a su vez constituyen el Backlog del proyecto, se determina de manera formal los requerimientos por parte del Dueño del Producto.

Es importante considerar el alcance de manera más segmentada, considerando dicho alcance no como una totalidad para el proyecto, sino, como un segmento del mismo. Al inicio de cada sprint o iteración, el equipo de trabajo deberá determinar cuál es el número máximo de historias de usuario que entrarán en la iteración, con lo que se definiría el micro alcance. Tomando en cuenta lo anterior, no sería de mucha ayuda invertir tiempo en la creación de una Estructura de Desglose de Trabajo (Work Breakdown Structure en inglés), ya que esta división de actividades se tomaría directamente al inicio de cada iteración.

Ahí mismo, al inicio de cada iteración se definirán las actividades que compondrán la iteración y cuál sería la secuencia a seguir, todo de manera consensuada por parte del equipo de trabajo. También, se estimaría los recursos a utilizar por parte del equipo de trabajo, sean estos materiales o humanos. La estimación de la duración de las actividades se definiría antes de la primera iteración, cuando se establece formalmente la duración de cada Sprint. El cronograma de actividades quedaría en manos del equipo de trabajo autogestionado.

La siguiente tabla muestra la relación de salidas y entradas de PMBOK y en cuál momento de la iteración debe documentarse:

Proceso PMBOK	Documentación	Momento del Sprint Agile
Recopilación de Requisitos	Documentación de Requisitos	Al inicio del Proyecto, mediante las Historias de Usuarios.
Definir el Alcance	Enunciado del Alcance del Proyecto	Al inicio del Proyecto, mediante las Historias de Usuarios. Backlog
Definir las actividades	Lista de actividades	Definición de las actividades del Sprint
Desarrollar el cronograma	Cronograma del Proyecto	Definición de las actividades del Sprint
Planificar la calidad	Indicadores de Calidad	Definido por el Dueño del Producto mediante las pruebas de calidad que defina.
Planificar las Gestión de Riesgos*		

Table 1: Proceso de Planeación

3.3 Proceso de Ejecución

Parte de los inconvenientes que pueden presentar las metodologías tradicionales como la “Cascada” es que no se puede avanzar con una etapa, hasta concluir la

anterior. De tal manera, la fase de Ejecución de un proyecto no puede iniciarse hasta que todos los planes se hayan creado, los interesados están satisfechos con lo plasmado y se haya disponibilidad de ejecutar una parte del presupuesto. En contraposición, los métodos ágiles impulsan la productividad mediante las iteraciones, las cuales permiten tener en periodos tan cortos como dos semanas, un producto funcional, aunque no perfecto.

Siendo que el PMBOK es una guía de las mejores prácticas de gestión de proyecto, como tal, contiene muy poco acerca de cómo llevar a cabo realmente las actividades del proyecto. Su propósito es que los gestores implementen técnicas adecuadas de acuerdo al campo en el que se vayan a trabajar. Para ilustrar el caso se puede decir que existe una gran cantidad de información disponible acerca de cómo ejecutar los proyectos de construcción y esta no está detallada en el PMBOK. Corresponde a los Ingenieros Civiles y Arquitectos aplicar las técnicas de gestión ligadas a sus conocimientos técnicos en la materia.

Siguiendo la idea anterior, los proyectos de desarrollo de software deben emplear los procesos de ejecución más alineados a las necesidades del proyecto. Algunas técnicas a utilizar (no siendo las únicas) pueden ser:

1. **Desarrollar iterativamente:** construir software iterativo e incremental permite hacer frente a las necesidades del cliente. El desarrollo iterativo también ayuda a la reducción de los riesgos al permitir que los elementos del nuevo producto se prueben tan pronto se sale de las iteraciones, en lugar de esperar hasta la entrega final del producto donde ser más problemático descubrir problemas, habiendo poco (o ningún) tiempo para su resolución.

El desarrollo iterativo también ofrece muchas ventajas adicionales sobre los enfoques tradicionales, incluyendo: mejor visibilidad progreso, oportunidades de aprendizaje, y las opciones para principios de realización de beneficios a través de comunicados de interinos.

2. **Desarrollar indicadores útiles:** Durante el desarrollo del proyecto se deben evitar caer en la tentación de invertir – o gastar – tiempo creando indicadores que no aporten valor al proyecto tales como horas laboradas por programador, ya que, más bien las metodologías ágiles buscarán que de manera fija no se trabajen más de 40 o 44 horas semanales; o bien, cantidad de líneas de código escritas ya que tampoco brinda mayores beneficios o luz respecto al objetivo general del proyecto.

De igual manera, el indicador de satisfacción por parte del cliente y el factor de Calidad que es determinada por el dueño del producto son indicadores importantes a generar.

Por el contrario, indicadores como Funciones por Iteración Concluida o Tiempo Restante de la Iteración tienden a ser más útiles.

3. **Potenciar el trabajo del equipo:** el modelo tradicional para la asignación de trabajo es un problema para los proyectos de software que son difíciles de estimar y, a menudo contienen dependencias impredecibles. Los intentos de micro-administrar los desarrolladores mediante la asignación de tareas individuales son propensos a fallar. En su lugar, es mejor aprovechar la capacidad de las personas para gestionar la complejidad de la asistencia en la definición de las metas y objetivos de alto nivel y luego capacitar al equipo con la responsabilidad de la consecución de los objetivos.

En términos prácticos, esto significa que la participación de los representantes de los equipos de trabajo y de los interesados en la planificación de características es fundamental, y a continuación, se debe dejar que el equipo de trabajo seleccione y desarrolle las características del producto en esa iteración como mejor les parezca. El progreso general se puede seguir a través de la entrega de características completas al final de cada iteración, sin embargo debe ser el equipo el que se autogestione.

Dado que las metodologías ágiles se basan en un desarrollo iterativo, es por tanto, necesario gestionar la calidad en cada una de las iteraciones. Los equipos de trabajo deberán tener en cuenta que antes del cierre del Sprint se deberá revisar que se cumpla con los indicadores de calidad propuestos por el Dueño del Producto. Al final de la iteración, dicho Dueño del Producto será responsable de validar los indicadores citados respecto a la funcionalidad que se le está presentando.

Pues bien, para mapear adecuadamente el tema de la Fase de Ejecución como aparece en el PMBOK respecto a la mecánica de las Metodologías Ágiles, se recomienda mantener los siguientes entregables o salidas:

Proceso PMBOK	Documentación	Momento del Sprint Agile
Dirigir y Gestionar la Ejecución del Proyecto	Lista de Entregables Solicitud de Cambios	Antes del Inicio del Sprint Entre cada Sprint
Aseguramiento de la Calidad	Indicadores de Calidad Asignación de Tareas	Antes del Inicio del Sprint Antes del Inicio del Sprint
Adquirir el Equipo del Proyecto y Dirigir el Equipo del Proyecto	Calendario de Recursos	Antes del Inicio del Sprint
Distribuir la Información	Actualización de los Logros	Al final de Cada Sprint

Table 2: Proceso de Ejecución

3.4 Proceso de Monitoreo y Control

Ya que, tanto el tiempo y el costo (presupuesto) son dos de las tres aristas que todo proyecto tiene, y que junto con el alcance, son en sí el proyecto, el PMI establece una fuerte supervisión de estos elementos a la largo del proyecto, intentando que ninguno de los dos se quede corto o se sobrepase. En la otra acera, los métodos ágiles también se gestionan controlando el tiempo y el costo, sin embargo, al estar en un proyecto que muy probablemente cambie con el tiempo, tanto el costo como el tiempo, y por ende el alcance cambiarán de la mano de las necesidades de los interesados.

Con el propósito de mejorar la gestión del proyecto, el Control Integrado de Cambios se gestiona directamente en el Backlog del producto, evidenciando tales cambios. Ya que, según se indicó anteriormente, la Gestión del Alcance se realizará al inicio de cada Iteración, no existiría mayor conflicto respecto al alcance. De manera similar, el cronograma se eximiría de control ya que, de manera predeterminada, se conoce que la duración de la iteración variará de 2 a 4 semanas, por lo que representará un esfuerzo menos por realizar. Finalmente, la Gestión de la Calidad irá de la mano con cada entrega parcial de producto al final de cada iteración, por lo que se recomienda formalizar la recepción por parte del dueño del producto.

3.5 Proceso de Cierre

Tal como se define propiamente un proyecto como un esfuerzo temporal, deberá tener un adecuado proceso de cierre. EL PMBOK busca dejar formalmente definido la aceptación de parte de los interesados respecto al producto que se les brindó. Los métodos Ágiles gestionan de una forma bastante diferente el tema de aceptación del producto, ya que dueño del producto, en representación directa de los interesados brindan esa aceptación del producto en porciones más pequeña a lo largo del proyecto al finalizar cada iteración.

En este punto se refleja por qué se puede obtener una mayor satisfacción por parte de los clientes a la hora de desarrollar un producto de software con metodologías ágiles, ya que no es lo mismo presentar una solución de software ya terminada y comenzar a explicar punto por punto, a que el cliente vea funcionando e producto, parte por parte, cada dos semanas.

4 Conclusiones

La transición de los Modelos Tradicionales de Gestión de Proyectos al desarrollo de software mediante las Metodologías Ágiles es complicada debido a las diferencias inherentes entre los dos enfoques. Hay varios conceptos que son similares y se pueden asignar de uno a otro, sin embargo pareciera que hay puntos más complicados de unificar, como el énfasis en la planificación por adelantado, que

son fundamentalmente diferentes, y por lo tanto difícil de conciliar entre los dos.

Sin embargo, como se ha expuesto, se puede lograr – al menos en el papel – obtener los beneficios de la robustez documental que ofrece el PMBOK, sin renunciar a la flexibilidad y adaptabilidad de las Metodologías Ágiles. Aún más, se podría inferir que, a pesar de que el presente trabajo versó sobre cómo implementar aspectos de una metodología tradicional como PMBOK en un modelo ya maduro de Gestión de Proyectos basado en Agile, perfectamente se podría buscar cómo integrar las metodologías ágiles en proyectos con ciclos de vida iterativos que no sean precisamente de software y que están gestionados a la medida del PMBOK.

Respecto a esta guía, se aplica la misma cláusula de restricción que brinda PMBOK: no se trata de una guía formal, sino una serie de prácticas, en cuyo caso, será responsabilidad del gestor del proyecto determinar su usabilidad y en qué proporción.

References

- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *Internation Journal of Engineering & Technology*, 2(5). pages 2
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). *Manifesto for agile software development*. Retrieved from <http://www.agilemanifesto.org/> pages 3
- Griffiths, M. (2004). Using Agile Alongside the PMBOK. *2004 PMI Global Congress Proceedings*, 5, 1–105. pages 3
- Layton, M. C. (2012). *Agile Project Management For Dummies* (M. Enochs, K. Feltman, & L. Michael, Eds.). New Jersey: John Wiley & Sons, Inc. Retrieved from http://books.google.co.uk/books?id=hasLwk{_}BAvEC pages 4
- Leffingwell, D. (2011). *Agile software requirements : lean requirements practices for teams, programs, and the enterprise*. Pearson Education, Inc. pages 6
- Project Management Institute. (2013). *Fundamentos para la dirección de proyectos (Guía del PMBOK)* (Quinta Edi ed.; PMI Publications, Ed.). Pennsylvania, USA. pages 2, 9
- Project Management Institute. (2014). *2013 AnnualReport* (Tech. Rep.). Retrieved from <https://www.pmi.org/About-Us/{~}/media/PDF/Publications/PMI-2013-Annual-Report-Web.ashx> pages 8
- Sharma, S., Sarkar, D., & Gupta, D. (2012). Agile Processes and Methodologies : A Conceptual Study. *International Journal on computer science and Engineering*, 4(05), 892–899. pages 7
- The Standish Group. (2005). The Standish group: the chaos report. *Project Smart*, 16. pages 2

