ANÁLISIS DE LA APLICACIÓN DE LA METODOLOGÍA SCRUM COMO COMPLEMENTO DE LAS METODOLOGÍAS DEL PMI PARA EL CONTROL DE PROYECTOS DE DESARROLLO DE SOFTWARE.

SCRUM METHODOLOGY APPLICATION ANALYSIS AS PMI METHODOLOGIES COMPLEMENT FOR PROJECTS CONTROL IN SOFTWARE DEVELOPMENT

Rafael Andrés Amézquita Mejía
Ing. En Multimedia
Facultad de Ingeniería
Investigador Gerencia Integral de Proyectos,
Universidad Militar Nueva Granada
Bogotá, Colombia
rafael.amt@outlook.com

RESUMEN

Se dan a conocer los conceptos básicos sobre el control del tiempo y el ciclo de vida de un proyecto según los planteamientos del PMI (Project Management Institute), para plantearlos como el soporte de metodologías como SCRUM, la cual replantea la manera en que se controlan los proyectos enfocados al desarrollo de software, tomando como premisa los constantes cambios que se presentan en estos proyectos.

ABSTRACT

The article introduce the basic concepts about time management and projects lifecycle, following the Project Management Institute (PMI) approaches as a support to the SCRUM methodology, which rethink the way how the software development projects are being controlled taking into account the frequent changes inside themselves.

Palabras Clave: scrum, PMI, management, gerencia, control, proyectos.
Keywords: scrum, PMI, management, gerencia, control, proyectos.

INTRODUCCIÓN

Dentro de la gerencia de los proyectos de software existen diferentes metodologías que permiten el control de los mismos, dentro de ellas se encuentran SCRUM y las propuestas por el Project Management Institute. De acuerdo a lo anterior, se tiene que, SCRUM plantea un pequeña modificación de las fases de los proyectos, volviéndolas actividades dentro de paquetes llamados entregables, que serán gestionados por un SCRUM MASTER y un equipo de desarrollo que a su vez se encargara de enfrentar cada riesgo que se presente.

Por otro lado, se tiene que, con las otras metodologías expuestas por el PMI, usualmente están hechas para periodos de tiempo muy extensos y en donde la mayoría de elementos con los cuales se trabaja no cambian con la frecuencia ni la intensidad con la que lo hacen en un proyecto de software. Dado esto, se tiene un escenario lleno

de riesgos en el que frecuentemente se vuelven realidad, y solo al final se de la fase de desarrollo se pretenden resolver.

De acuerdo a lo anterior, se busca encontrar beneficios, cuyo objetivo sea tener un producto altamente funcional que brinde satisfacción tanto al cliente como a los miembros del equipo.

1. MATERIALES Y MÉTODOS

Con el fin de evidenciar la aplicabilidad de estas metodologías se realizará la recolección de información mediante un estudio exploratorio y descriptivo, en el sentido que se consultará a expertos en la gerencia de proyectos dentro de la industria del desarrollo de software, y a su vez se exploraran fuentes secundarias de información para aclarar conceptos y visualizar como complementa una metodología a la otra.

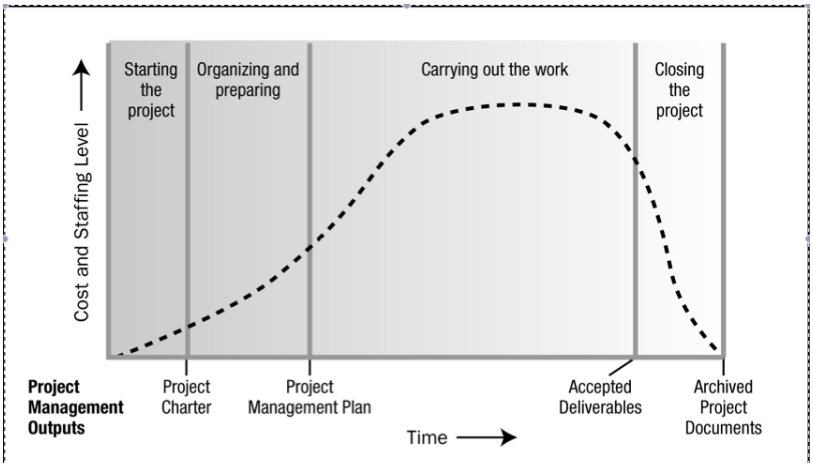
Para entender lo anterior, se describirán los conceptos básicos, sobre el ciclo de vida de un proyecto general, según el *PMBOK (Project Management Body of Knowledge)*. Luego una descripción elemental, sobre lo que plantea la metodología *SCRUM.* Con esto, se podrán narrar, los problemas dentro de la industria del software, cuando se hace gerencia de proyectos únicamente bajo lo propuesto por el *PMBOK,* y como se solucionan cuando las metodologías se complementan con los planteamientos de *SCRUM.*

2. MARCO CONCEPTUAL

2.1 Conceptos básicos del ciclo de vida del desarrollo de proyectos según el PMI.

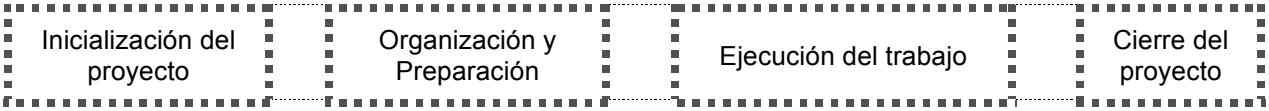
El ciclo de vida de un proyectos esta compuesto por una serie de fases que varían dependiendo las necesidades de los involucrados, cada una de estas fases esta estructurada por una serie de aspectos que se definen en el entorno en donde se desarrolla el proyecto, por quienes lo manejan o por una tecnología que se use, sin embargo el PMI, brinda un marco conceptual generalizado de lo que se considera una correcta organización para el ciclo de vida. Project Management Institute, Inc. [1]:

Figura 1. Niveles típicos de costo y dotación de personal durante el ciclo de vida del proyecto.



Fuente: PMBOK Body Of Knowledge 4th Edition. Capítulo 2.

Figura 2: ciclo de vida de un proyecto



Fuente: Autor

Dentro del ciclo de vida del proyecto, un gerente de proyectos puede determinar que otros ciclos se pueden manejar para ejercer un mejor control sobre el desarrollo del producto, estos subciclos o fases pueden variar dependiendo las necesidades del producto y el entorno de trabajo.

2.2 Introducción a *Scrum*: conceptos básicos.

Para definir *Scrum*, es necesario especificar primero lo que no es: no es una técnica o un proceso, en lugar de esto, es un marco de trabajo en el que se pueden aplicar procesos y técnicas para el desarrollo de nuevos productos, el cual introduce un ciclo de retroalimentación, cuya meta es la construcción de practicas que sirvan para el desarrollo de productos complejos, que en este caso son los que están dentro del desarrollo de software. También se basa en las premisas expuestas por el "manifiesto ágil", como son:

" 1. Las interacciones y los individuos, por encima de los procesos y las herramientas.
2. Software funcional, por encima de extensa documentación.
3. Colaboración con el cliente, por encima de contratos y negociaciones.
4. Respuesta al cambio, por encima del plan de trabajo.

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda." Kent Beck et al [2].

Con lo anterior, la manera en que *Scrum* simplifica el proceso de desarrollo de software, es introduciendo el concepto de Sprint, lo que significa tomar las fases:

Figura 3. Proceso de desarrollo de software

Fuente: Autor

dentro de cortos periodos de tiempo (los periodos pueden variar, dependiendo las negociaciones realizadas con los clientes), dentro de los cuales se define un producto entregable potencialmente funcional, que cumpla de manera parcial con las características requeridas por el cliente, con lo que, en la medida en que se cumpla con una cantidad de *Sprints* determinada, el software se va completando hasta terminar con la aprobación del cliente.

2.2.1 Roles

El equipo con el cual se conforma el *Scrum*, esta dividido entre: el *ScrumMaster*, el *Product Owner*, y final mente el equipo (*Team*); Scrum Alliance [3]. Dicho esto el *ScrumMaster* trabaja con los cliente y la gerencia para informar al *Product Owner* como manejar y optimizar el valor que se pretende generar.

2.2.2 ScrumMaster

Es considerado como el rol más importante, ya que, acata como intermediario entre el dueño del producto y el equipo de trabajo, filtrando toda las actividades pertinentes al alcance de cada entregable y reorganizando los requerimientos adicionales, evitando de esta manera la desorientación del equipo de desarrollo; Scrum Alliance [3].

El *ScrumMaster* enseña y lidera al equipo en la tarea de ser más productivo y en generar productos de mayor calidad, sin embargo, es importante resaltar que el *ScrumMaster* no gerencia al equipo de trabajo, este es auto regulado.

2.2.3 Product Owner

Es la persona encargada de controlar la realización del producto y del valor que este tiene, ya que también es la representación del equipo de trabajo o el responsable del proyecto ante el cliente; Scrum Alliance [3].

Es el encargado de mantener el flujo de trabajo y dar a conocer al equipo, cuales actividades son las de mayor prioridad. Es el único con la autoridad para cambiar la priorización de las actividades, pero la principal característica de este rol es la comunicación con los interesados en el proyecto (los *Stakeholders*).

2.2.4 Team

Esta conformado por los desarrolladores, diseñadores y todos los involucrados en la construcción del software; Scrum Alliance [3]. Esta diseñado para optimizar flexibilidad y productividad, son auto organizados, multidisciplinarios y trabajan por iteraciones (por *sprints*). Es el que convierte toda la lista de actividades en un producto potencialmente funcional con cada Sprint.

A pesar de que cada miembro del equipo tiene una especialidad, es decir, uno es el administrador de las bases de datos, otro es arquitecto de software, diseñador de

interfaces de usuario, etc, todos deben compartir una misma habilidad, esta es, la capacidad de convertir su trabajo en un producto usable.

2.3 Puntos de Control - Time Boxes

Figura 4. Puntos de control establecidos por el *Scrum*

Fuente: Autor

2.3.1 Release Planning Meeting

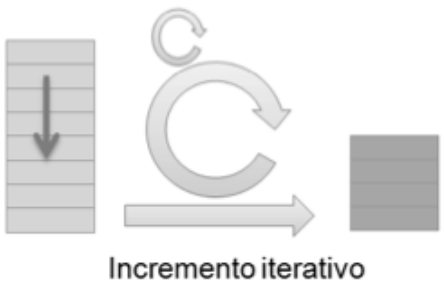
El propósito de estas reuniones es lograr con el equipo, establecer metas claras y un plan que el mismo equipo y el resto de la organización puedan entender y comunicar; Scrum Alliance [3]. Básicamente el *Release planning meeting* responde a las siguientes preguntas:

- ¿Como tornar la visión en un producto exitoso, de la mejor manera posible?.
- ¿Como se puede conocer el punto de satisfacción del cliente?
- ¿Como se puede conocer el punto de retorno de la inversión por parte o para el cliente?

2.3.2 Sprint

El *Sprint* es una iteración, es un tiempo planeado, durante el cual el *ScrumMaster*, se asegura que no exista ningún cambio que pueda llegar a afectar los objetivos propuestos para un determinado periodo de tiempo; Scrum Alliance [3]. Cada proyecto consiste en la definición de lo que se va a construir, un plan para construirlo, el proceso o trabajo con el que se construye, y finalmente el producto resultante (este puede ser el producto objetivo o bien una parte total funcional del producto general).

Figura 5. Visualización del *Sprint*

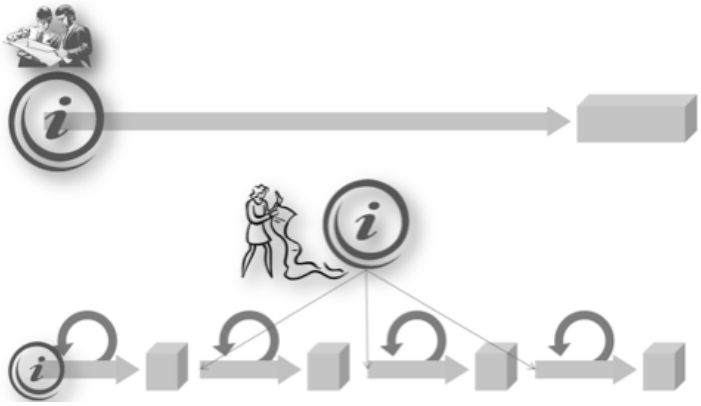


Fuente: Gestión de proyectos *Scrum* Manager.

2.3.3 Sprint Planning Meeting

Es la reunión en donde el *Sprint* es planeado, usualmente su duración es de ocho horas y los *Sprints* que se planean tienen una duración de un mes; Scrum Alliance [3]. Esta reunión se divide en dos partes de cuatro horas, en la primera se tratan temas de que metas se tienen que alcanzar y cuales serian las prioridades, y en la segunda, se encuentra el equipo de trabajo y se planea como se va a llevar a cavo la construcción del producto, teniendo en cuenta las prioridades y la lista de actividades.

Figura 6. visualización de los ciclos iterativos (*Sprints*) para lograr un producto.



Fuente: Gestión de proyectos *Scrum* Manager.

2.3.4 Sprint Review

Es una reunión usualmente a final del mes (teniendo en cuenta que se ha planeado un sprint con esa duración), de cuatro horas, en donde, los miembros del equipo y los *stakeholders*, revisan lo que se realizó, se pactan correcciones o mejoras, y se define que puede entrar para el siguiente *Sprint*; Scrum Alliance [3].

2.3.5 Sprint Retrospective

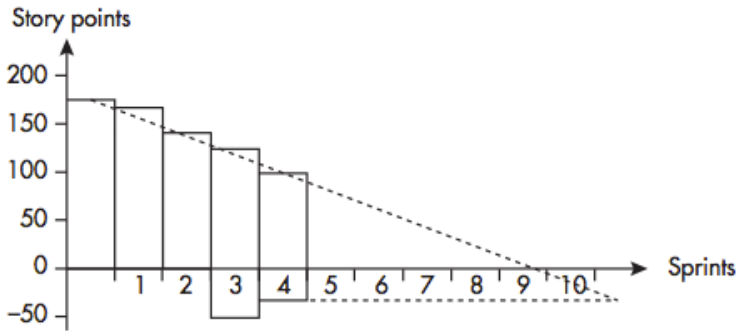
Después de la reuniones con los *stakeholders,* los miembros del equipo se reúnen, aproximadamente por tres horas, en donde el *ScrumMaster*, discute con su equipo de desarrollo, mejores practicas y alternativas para hacer mas eficaz el siguiente *Sprint*; Scrum Alliance [3].

2.3.6 Daily Scrum

Es una reunión de máximo quince minutos en donde se encuentra el equipo de desarrollo con el propósito de mejorar la comunicación, solucionar posibles errores que estén afectando la continuación del proyecto; Scrum Alliance [3]. Básicamente se responden a las siguientes preguntas:

- ¿Que actividades se realizaron desde la ultima reunión?
- ¿Cuales son las actividades que se van a realizar para la siguiente reunión?
- ¿Que posibles obstáculos se pueden presentar para la realización de estas actividades?

Figura 7. Flujo de trabajo, dado por cada *sprint*



Fuente: Agile Product Management With Scrum. Capitulo 5.

3. RESULTADOS Y ANÁLISIS

3.1. Entrevista

3.1.1 ¿Por que *Scrum* y no cualquier otra metodología ágil?

Puede que cualquier metodología que este reverenciada como ágil pueda servir, pero, algo que encontré muy útil y que de hecho me ha servido desde que lo implemente, es que el mismo equipo es quien responde ante las variantes del día a día, y uno como gerente ya no se expone ante un cliente a decir cosas que uno no sabe.

3.1.2 ¿Una diferencia entre un gerente que aplique solo las metodologías del PMI y el que implementa *Scrum*?

Con mi experiencia, en este negocio, nunca he tenido buenos resultados con gerentes certificados como PMP, tienen cosas muy bonitas en el papel, pero en la realidad todo es un desastre.

3.1.3 ¿Que es lo mejor de *Scrum*?

La forma en que se acota el producto, con eso el equipo como el cliente, siempre están motivados, por que ese demo que se entrega con los *sprints,* da la seguridad que las cosas tienen buena marcha y si se hacen cambios el cliente es consiente, que lo que quería inicialmente ya se hizo.

3.2 Principales problemas en el desarrollo de software, con las metodologías del PMI.

Una de las muchas situaciones que se presentan dentro de los proyectos que involucran desarrollo de software, es el control inadecuado del proceso, es decir, no se tiene la diferenciación que la labor de pensar para la elaboración de código fuente, es una actividad totalmente diferente a un proceso repetitivo cuantificable, con resultados predecibles.

Para hacer más claro lo anterior, se podrían comparar, los proyectos de construcción de obras civiles y los proyectos de construcción de software. Estos dos proyectos requieren las mismas fases que cualquier otro proyecto: inicialización, organización y preparación, ejecución del trabajado y cierre del proyecto. Ambos necesitan que la mano de obra, este alineada con un cronograma, cuyas actividades estén monitoreadas por un encargado y ambos necesitan exponer su progreso ante un cliente. Sin embargo, las diferencias comienzan a ser notables en el punto de ejecución del trabajo, ya que, la construcción de una obra civil, parte de labores repetitivas, con una duración bien conocida, dirigida por un maestro de obra, quien se rige por unos planos previamente realizados, pero, la construcción de software no se puede realizar de esta manera, y con esto se especifica que el punto más simple que se debe tomar en cuenta es que, en esta ocasión, la mano de obra es calificada y profesional, seguido de la evidente complejidad de las actividades dentro de los cronogramas, esto quiere decir que, ninguna actividad llega a ser repetitiva y que un desarrollador, no toma el mismo tiempo para resolverla que otro, así tengan el mismo nivel de conocimiento.

Un problema, muy frecuente, que se presenta por parte del gerente de proyectos, es este trato a su personal descrito anteriormente; se cae en el error de pensar que las actividades dentro del cronograma van a durar lo mismo, sin importar quien las resuelva. Profundizando un poco en esto, incurrir en este pensamiento, no solo lleva al fracaso de cualquier desarrollo de software, si no que, en determinado momento, satura al trabajador, quien por su agotamiento, no puede garantizar un producto de calidad.

Figura 8. Cuadro comparativo entre la industria de la construcción y la industria del software

Actividades y requerimientos	Proyectos en la industria de la construcción	Proyectos en la industria del desarrollo de software
Mano de obra regida por un cronograma de trabajo	*x*	*x*
Monitoreo y control en la ejecución de las actividades	*x*	*x*

Actividades y requerimientos	Proyectos en la industria de la construcción	Proyectos en la industria del desarrollo de software
Exposición del progreso ante un ente regulatorio o cliente	*x*	*x*
Tiempo de ejecución de una actividad determinado	*x*	
La ejecución de las actividades operativas puede ser por personal no calificado	*x*	
El tiempo de ejecución de las actividades es indiferente de quien las realice	*x*	

Fuente: Autor.

Otro problema, bien conocido, en esta industria, se presenta durante la organización y preparación del proyecto, más específicamente en la planeación y construcción del cronograma, ya que se parten de históricos para estimar las duraciones de las actividades, sin embargo, esto es otro error, ya que, como se dijo anteriormente, ningún trabajador se demora lo mismo que otro, haciendo exactamente lo mismo, de hecho, un mismo trabajador, no se demorará lo mismo haciendo una actividad en dos momentos diferentes, es decir, para el desarrollador “a” hoy toma una semana realizar un modelo de datos, pero mañana puede tomar dos semanas o media semana.

Siguiendo con los problemas, también se encuentra otra falla común dentro de la gerencia de proyectos, aunque esta se sale de los planteamientos del *PMBOK,* hace parte de la cultura organizacional de muchas compañías, pequeñas y grandes, y esto es, pensar que delegar un proyecto a los empleados significa tener un producto terminado, sin la necesidad de una supervisión y control adecuados, es decir, un gerente bajo ningún motivo, puede pretender que su producto se va a hacer sin requerir su presencia. Si bien es cierto que, un equipo de desarrollo, puede lograr (y en muchos casos lo hace) el producto pretendido, sin requerir la presencia de un gerente, incurrir en esto, representa un riesgo muy alto, ya que en esta industria, el cambio, es un elemento natural que se vive en el día a día. Un ejemplo de esto, es que hoy se tiene un equipo con la mejor disposición y motivación para construir el producto, pero al día siguiente se encuentran con un obstáculo que requirió el cambio de la arquitectura de todo el sistema, por lo que el tiempo planeado se multiplica, y si el gerente no está para ver esto, cuando se decida por revisar el avance, se encontrará con que solo se ha hecho la mitad de lo que se requirió, pero será demasiado tarde, para lograr intervenir y mitigar el efecto negativo que esto trae al cliente.

3.3 ¿Como se complementan las metodologías propuestas por el PMI y SCRUM?.

En la practica un equipo empresarial, puede lograr llevar un proyecto de software exitosamente solo implementando *scrum,* sin embargo cuando se usan los

fundamentos descritos en el *PMBOK,* la evolución del equipo será mucho mayor, ya que aprovecharan todos procedimientos realizados en las diferentes areas del conocimiento, para mejorar la calidad de sus productos en el presente y futuro.

Con *scrum,* se puede llegar a tener una aproximación a la gestión del proyecto, pero son las metodologías que se plasman en el *PMI* las que abarcan todos los detalles del mismo. Dicho esto, se puede decir que aunque los planteamientos del *PMBOK,* son la guía para entender el proyecto, se enfocan más en su gestión que en el proyecto como tal, es decir, el enfoque de *scrum*, es más al valor agregado y a la calidad del producto, que a la documentación y la gestión de la generalidad del mismo.

Un gerente de proyectos que pretenda llevar a cavo la construcción de un software, y desee implementar lo aprendido del *PMBOK* se puede beneficiar de *scrum* desde la planeación del proyecto, ya que al proyectar unos “demos” (que realmente son los entregables), de manera periódica e iterativa, logrará que con cada iteración (*sprint*) se disminuya significativamente el trabajo, y le va a otorgar al cliente la visualización de su producto a medida que este se va construyendo.

Viendo el valor ganado en los “demos”, durante la ejecución del trabajo, se puede ver otro complemento, y es la manera en que se trata el riesgo entre todos los integrantes, ya que con *scrum,* el equipo de desarrollo gestiona una constante comunicación, con la que, se resuelven los obstáculos que se presentan durante este proceso, casi en tiempo real, es decir, si un desarrollador tiene un inconveniente en terminar una actividad, este la comunica en el *daily scrum,* en donde sus colegas le colaboran y solucionan el inconveniente.

Otro complemento por parte de *scrum*, que realmente, es más, una solución a un inconveniente dentro del planteamiento del *PMI,* es la manera en que se estiman los tiempos de ejecución de cada actividad, ya que no se parten de históricos, sino, de un consenso con el equipo de desarrollo en donde es, este mismo quien indica cuanto se va a demorar, lo que genera estimaciones más aproximadas y menos vulnerables de ser incumplidas.

3.4 Principales aportes de SCRUM a proyectos del desarrollo de software.

Compañías como Amazon, Adobe, entre otras, usan *Scrum* como la metodología mas idónea para gestionar sus proyectos, por sus principales beneficios, uno de ellos presentado por la compañía Adobe, es el hecho de poder alcanzar la meta de construir, lograr y validar los detalles del proceso, todo el tiempo, ya que con esto, lograron evidencias de posibles bloqueos a la realización progresiva de sus metas.

Lo anterior y datos recolectados entre diferentes gerentes y líderes técnicos, muestran que los principales aportes de usar esta metodología en los diferentes proyectos, son:

1. Validación regular de las expectativas del cliente.
2. El logro de resultados anticipados.

3. La adaptación a las diferentes variables que se presenten, tanto positivas como negativas.
4. Retorno de la inversión por parte del cliente.
5. Prevención de riesgos con el suficiente tiempo de control.
6. Una mejor comunicación entre el cliente y el equipo.
7. El logro de un equipo motivado.

Esto ultimo toma importancia, en el sentido que cada integrante, mantiene con la satisfacción de completar de manera parcial el producto deseado, situación que en otras metodologías es difícil de observar ya que no es tan evidente el entregable, como si lo es en la culminación de cada *Sprint.*

Con *scrum* se entiende que el cambio es un elemento natural a los proyectos de desarrollo de software, se logra con el concepto de *sprint* lidiar con los ciclos del trabajo sin interrupciones, acotando el alcance con cada uno, en el manejo del riesgo, el equipo reconoce los problemas y estos mismos se resuelven en tiempo real, se hace uso del conocimiento experiencial o heurístico para la presentación de estimados en los cronogramas, es decir, que se tiene en cuenta al equipo para tomar estas medidas.

CONCLUSIONES

- Se visualizó, el hecho, que acotar el alcance del proyecto y distribuirlo mediante entregas parciales, potencialmente funcionales, da valor agregado al producto ante el cliente.

- Se evidenció, que la comunicación diaria entre los miembros del equipo, durante la ejecución del trabajo, puede disminuir los riesgos de retrasos en las actividades propuestas en el cronograma.

- Se expuso, como por medio de SCRUM, se pueden potencializar las metodologías propuestas en el PMBOK, con el fin, de tener un producto de calidad y un equipo motivado.

- Se recomienda, siempre que se pretenda, construir un cronograma, no usar históricos, sino, basarse en la experiencia de cada uno de los integrantes del equipo.

- En el momento, en que un proyecto comienza, se recomienda, mantener un control, en lo posible diario, y con eso, prevenir los riesgos que se suelen presentar en esta industria.

- En la planeación y organización de las actividades, se recomienda destinar todo un día si es necesario, con los integrantes del equipo, con esto, se puede tener mayor seguridad sobre la construcción del producto, tanto por parte de la gerencia como por parte del equipo de desarrollo.

AGRADECIMIENTOS

Federico Ortega Nieto
Project Manager Senior.

REFERENCIAS BIBLIOGRÁFICAS

[1] Project Management Institute, Inc. (2008). Fundamentos para la dirección de proyectos (Guía del PMBOK) Cuarta edición. Global Standard.

[2] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas (2001). Manifiesto por el Desarrollo Ágil de Software. En: http://agilemanifesto.org/iso/es/

[3] Scrum Alliance; (2009). Scrum Master Guide.

[4] Jose Vázques Sáchez (2012). ¿Es posible mapear las prácticas de PMI a prácticas Agile/Scrum?. En: http://www.gestiondeproyectosit.es/blogit/2012/04/from-pmi-to-scrum-part-1/.

[5] Roman Pichler (2010). Agile Product Management with Scrum. Addison Wesley.

[6] Israel Gat, Michael Coté (2009). Scrum at Amazon – Guest Post by Alan Atlas. En:http://theagileexecutive.com/2009/07/20/scrum-at-amazon-guest-post-by-alan-atlas/.

[7] Adobe Systems Incorporated (2009). Adobe Premiere Pro Scrum Adoption – How an agile approach enabled success in a hyper-competitive landscape. En: http://blogs.adobe.com/agile/files/2012/08/Adobe-Premiere-Pro-Scrum-Adoption-How-an-agile-approach-enabled-success-in-a-hyper-competitive-landscape-.pdf