

Gestión de cambios en proyectos de Software. Change management in software projects.

Jhon Fredy Sandoval Llantén¹
jhon.sandoval02@usc.edu.co

Erick Mauricio Riascos Satizabal²
erick.riascos00@usc.edu.co

Orlando Ospina Lopez, M.Sc
oros@usc.edu.co

Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería de Sistemas (1)
Universidad Santiago de Cali, Facultad de Ingeniería, Programa de Ingeniería de Sistemas (2)

Resumen

El presente artículo tiene como propósito realizar una revisión bibliográfica acerca de la gestión de cambio como parte esencial de un proyecto de software. Los cambios son actividades espontáneas que pueden surgir en cualquier fase del desarrollo de un proyecto y puede llegar a ser un proceso desafiante si no se tiene una gestión idónea y pertinente para dicho requerimiento, generando a menudo un riesgo que puede poner en peligro la integridad del proyecto en términos de costos, tiempo y funcionalidad, por ello es importante identificar, planificar, implementar y evaluar los diferentes cambios que pueden surgir en el proceso. Permitiendo así, realizar un proceso organizado, que garantice al usuario la minimización de errores y obtener el éxito del proyecto. En este artículo hemos listado y comparado algunas de las metodologías usadas en la actualidad, de las cuales concluimos que para su éxito se necesita flexibilidad al cambio, rápida toma de decisiones y un canal de comunicación óptimo.

Palabras Clave: Gestión de cambios, software, metodologías ágiles.

Abstract

The purpose of this article is to carry out a bibliographic review about change management as an essential part of a software project. Changes are spontaneous activities that can arise at any stage of the development of a project and can become a challenging process if it is not properly and appropriately managed for said requirement, often generating a risk that may jeopardize the integrity of the project. In terms of cost, time and functionality, it is therefore important to identify, determine, plan, implement and evaluate the different changes that can be surgical in the process. This allowing an organized process to be carried out, which guarantees the user the minimization of errors and the success of the project. In this article we have listed and compared some of the methodologies used today, from which we conclude that for success you need flexibility to change, fast decision making and an optimal communication channel.

Keywords: Change management, software, agile methodologies.

1. INTRODUCCIÓN

La gestión de proyectos es una secuencia de actividades a las que se les hace seguimiento y control, para alcanzar un objetivo específico, administrando y planificando los recursos asignados para la ejecución del mismo, de una forma que se pueda cumplir con el alcance en el tiempo estimado, llevando con éxito cada una de una las metas y propósitos. Estas actividades están agrupadas por cinco fases que componen el ciclo de vida del proyecto en su totalidad, el motivo primordial de que se dividen en fases es para facilitar la gestión y mejorar el control de una forma que se pueda mantener alineado con los objetivos establecidos, en ejemplo una de las actividades es la gestión de cambio. Así mismo podemos agregar que hay muchas formas de realizar la gestión de cambios, por tal motivo han surgido muchas metodologías que abarcan esta gestión,

sin embargo, cuando se trata de desarrollo de software no muchas tienen éxito.

Ahora bien, los proyectos tienden a tener cambios, lo que resulta en una modificación de los recursos que se disponen, el desarrollo de software al ser un producto con características abstractas tiende a tener más cambios que los proyectos de otras áreas (Chiapello & Greiner. 2018), lo que conlleva a tener más riesgos de fracaso. La gestión de cambio es vital al momento de gestionar un proyecto, considerando que no solo permite prever e identificar las modificaciones a tiempo, sino también controlar estos requerimientos, garantizar que se realicen de la manera más idónea y brindar un reporte real de estos a todos los involucrados en el proyecto, auditar el proceso y asegurarse de mantener una calidad alta y limpia frente a los diferentes mecanismos de cambios, permitiendo disminuir así el consumo de tiempo, presupuesto y recursos que se usarían para mitigarlos.

Muchos de los problemas que se suelen asociar a la gestión de proyectos tienen que ver con el cambio (González, 2018). Por tal motivo, son una de las mayores causas de que estos fracasen, debido a que al no estar planeados pueden atraer riesgos a la ejecución, los cuales deben ser mitigados para evitar gastos de presupuesto, tiempo y recursos. En proyectos de desarrollo de software es incluso más drástico, debido a que en estos es más frecuente la presencia de cambios; sea por fallas que se presentaron al momento de realizar pruebas en el código, se descubrió que se puede realizar un paquete de una forma más efectiva/eficiente, se realizó un mejor entendimiento de las necesidades del cliente o simplemente por un cambio de opinión del cliente.

De ahí que en este artículo se pretenda exponer algunas de las metodologías en el desarrollo de software utilizadas en la actualidad que varían en técnicas para abordar la planificación de un buen proyecto: Scrum, Metodología CMOD integrado con EVM, Extreme - Programming, Kanban, Scrumban. El objetivo de este trabajo es investigar acerca de las técnicas de preparación que existen en cada una de las metodologías mencionadas, teniendo en cuenta que esta fase de gestión de proyectos tiene una gran trascendencia para la buena marcha del mismo y puede ser esencial, debido a que en ocasiones esta parte es menospreciada lo cual puede llevar el proyecto al fracaso.

2. MARCO CONCEPTUAL

2.1. ¿Qué es la gestión de proyectos?

La gestión de proyectos es un conjunto de actividades que se realizan para cumplir o alcanzar un objetivo específico y bien definido como desarrollar un producto nuevo, realizar una campaña publicitaria, implementar nuevos servicios, realizar alguna mejora de un producto o servicio, entre otros (Carranza. 2016). Este esfuerzo debe realizarse durante un periodo de tiempo definido (con un inicio y un final), en el cual se debe planear y gestionar algunos recursos tales como: el presupuesto del proyecto, las materias primas que se van a utilizar, las herramientas y dispositivos, el recurso humano asignado a las actividades, entre otros. Es importante que al momento de definir el objetivo se deba tener en cuenta las necesidades del cliente o la empresa, ya que son estos los que al final darán la aprobación del éxito del proyecto e incluso la aprobación para aumentar los presupuestos y los recursos (Pmbok. 2017).

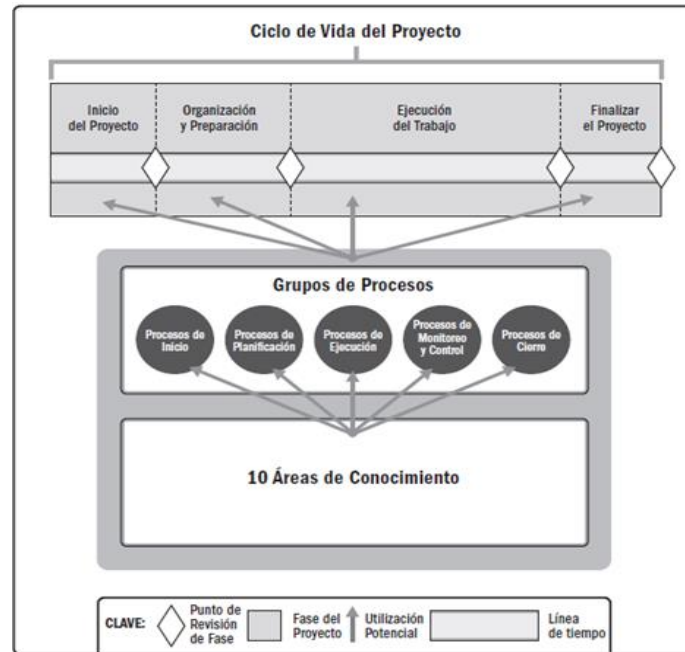
Al realizar una gestión de proyecto es muy importante tener buenas métricas en todo el ciclo de vida de este, debido a que la falta de estas puede impedir que tengamos un buen control y monitoreo del proyecto, lo que conlleva a no poder visualizar el avance, tener mala optimización del tiempo, presenciar una mala gestión de los recursos y/o el presupuesto y ganar poca conformidad por parte de los clientes (Ahmad; Hussain; Rehman; Qamar & Afzal. 2015).

El ciclo de vida de un proyecto está compuesto por cinco fases de proceso, los cuales componen el proyecto en su totalidad como lo muestra la figura 1. Es decir, desde el inicio hasta el fin, estas fases están compuestas por actividades que se deben realizar cuidadosamente para cumplir con el objetivo del proyecto. Una fase está compuesta por una o varias actividades las cuales al terminarse pueden generar un entregable, los cuales al ser medidos sirven como un indicador del desempeño del trabajo (Pmbok, 2017), a continuación, se exponen las 5 fases de los procesos:

- Proceso de inicio.

- Proceso de planificación.
- Proceso de ejecución.
- Proceso de monitoreo y control.
- Proceso de cierre.

Figura 1. Interrelación entre los Componentes Clave de los Proyectos



Fuente: Adaptado de (Guía del Pmbok).

2.2. ¿En qué fase está la gestión de cambios?

La fase encargada de realizar los procesos de monitoreo y control del proyecto tiene como objetivo principal realizar un seguimiento del progreso de las actividades, analizar y regular el desempeño del proyecto para efectuar una retroalimentación constante tanto al grupo de trabajo como a los interesados y poder asegurar el éxito del proyecto. Además, esta fase nos permite hacer un pronóstico de la tendencia del proyecto por medio de sus indicadores de desempeño de igual forma predecir posibles cambios o riesgos que puedan surgir y realizar un plan o tomar decisiones tempranas para mitigar el impacto de estos (Pmbok, 2017).

Es en esta fase, donde se encuentra la gestión de cambio implícita en el control integrado de cambios, una premisa en la gestión de proyectos es: las cosas cambian (Huércano, 2014) es inevitable el cambio, por tal motivo es necesario tener el procedimiento, las técnicas, los métodos y las herramientas para controlar y administrar a las personas y los recursos para lograr un resultado deseado después de un cambio (Mogogole & Jokonya, 2018).

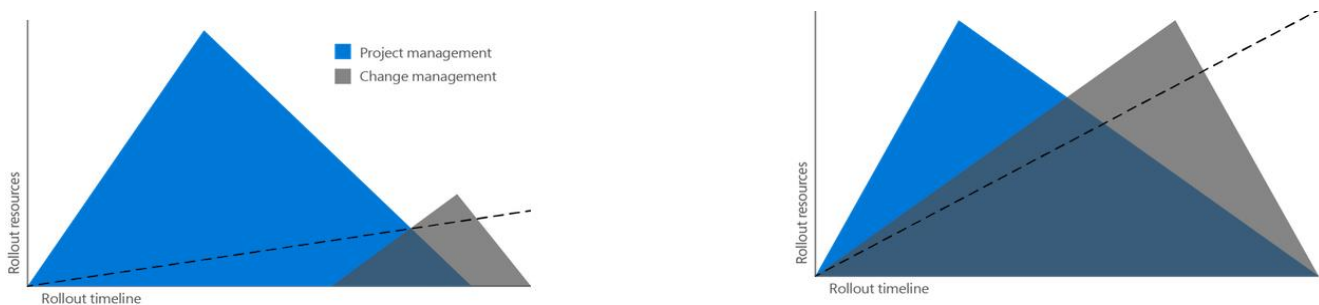
2.3. ¿Qué es la gestión de cambios?

La gestión del cambio es la actividad que tiene como objetivo minimizar los incidentes al planificar, analizar y evaluar los cambios que pueden surgir al desarrollar un proyecto, estos cambios pueden ser planeados o inesperados y se pueden presentar como un riesgo o una oportunidad de mejora. Los cambios pueden surgir por diferentes razones: por fallas que se presentaron al momento de realizar pruebas de código, se encontró una forma de hacer un paquete más eficiente/efectiva, se realizó un mejor entendimiento de las necesidades del cliente o por la modificación del alcance por petición del cliente o de los interesados (Dos Santos; Urdan & Mangini. 2016).

El cambio debe manejarse de manera conveniente y que requiere de un adecuado liderazgo para que se desarrolle de manera eficaz y con éxito (Villa Morales, 2019). La gestión de cambios es el proceso de lidiar con los cambios de requisitos que ocurren durante o después del desarrollo del software, Los cambios son inevitables y en muchas ocasiones difíciles de predecir, cuando un cambio surge de forma inesperada puede ser un gran problema o un riesgo debido a que al no estar planeado se debe gastar recursos que estaban contemplados para otras actividades y si no se cuenta con un presupuesto y recursos de reserva amplios se puede poner en riesgo el mismo éxito del proyecto (Ahmad; Hussain; Rehman; Qamar & Afzal. 2015).

La gestión de cambios es sumamente importante para lograr el éxito del proyecto, Microsoft nos presenta la diferencia del crecimiento de un proyecto de cuando se usa en todo el ciclo de vida a cuando se usa solamente en un momento (ver Figura 2), incluso la proponen como un indicador o causa del éxito (Montero. 2016; Motoa 2015), sin embargo, es una de las más omitidas.

Figura 2. Project management VS Change management.



Fuente: (Microsoft. 2018).

2.4. ¿Qué es la Gestión de cambios de requerimientos?

El cambio de requisitos en desarrollo de software es inevitable y puede ocurrir debido a cambios en los requisitos del usuario, una mayor comprensión de las necesidades de los interesados, la reestructuración organizacional del cliente y la disponibilidad de nuevas tecnologías. La gestión oportuna de estos cambios es vital para el desarrollo exitoso del software, que se puede lograr a través de un riguroso proceso de gestión de cambios de requisitos (Ali & Lai 2016).

Los requisitos de software a menudo no se establecen en concreto al comienzo de un proyecto de desarrollo de software; y los cambios en los requisitos se vuelven necesarios y, a veces, inevitables debido a los cambios en los requisitos del cliente y los cambios en las reglas comerciales y los entornos operativos; por lo tanto, el desarrollo de requisitos, que incluye cambios en los requisitos, es parte de un proceso de software. Investigaciones anteriores han demostrado que no gestionar bien los cambios en los requisitos de software es uno de los principales contribuyentes al fracaso del proyecto. (Shalinka, 2018).

Consideramos importante abordar este tema en el artículo debido a que los cambios no solo se presentan por modificaciones de presupuesto o cronograma, también incluyen en gran medida los cambios de los requisitos o modificación del alcance y saber o no cómo abordar esta gestión puede traer problemas o en su defecto mejorar enormemente el desarrollo del proyecto.

2.5. ¿Qué es el desarrollo global?

El Desarrollo Global de Software (GSD) es el que se realiza con miembros del equipo ubicados en todo el mundo y que desarrollan software comercialmente viable para empresas. Aunque presenta algunos problemas como la comunicación debió a diferencias de idioma del equipo y diferencias culturales, también tiene sus beneficios como el de reducir el costo del desarrollo de software ya que se puede realizar en países donde la mano de obra es más barata, acceso a recursos humanos calificados o permitir realizar actividades durante las 24 horas del día, y los 7 días de la semana debido a los múltiples sitios de trabajo. (Kamal; Zhang& Akbar, 2019).

3. CONTENIDO / DESARROLLO

Como lo hemos mencionado anteriormente, el objetivo de este artículo es exponer algunas de las metodologías para realizar la gestión de cambios centrándonos en proyectos de software, con el fin de compararlas y definir cuál es la más adecuada para los proyectos de software, para esto hemos seleccionado artículos limitándose solamente a los que se han publicado desde hace 5 años a la actualidad y que se centran en la gestión de cambios en proyectos de software o gestión de cambios de requisitos en proyectos de este mismo tipo.

A continuación, exponemos diferentes tipos de metodologías ágiles de desarrollo de software encontradas en diferentes artículos, que fueron totalmente seleccionadas todas aquellas que se centran en la gestión de cambio en proyectos de software, independientemente si utilizan como base otras metodologías antiguas, se complementan con otras o simplemente sean metodologías completamente nuevas.

3.1. Extreme - Programming (XP)

La programación extrema o eXtreme está especialmente diseñada para el desarrollo de software, XP es una metodología ágil centrada en la potenciación de las relaciones interpersonales, esta metodología tiene presente la realimentación continua entre el cliente y el equipo de desarrollo, el objetivo de XP es especialmente adecuado para proyectos con requisitos imprecisos y cambiantes por lo que es muy importante su marco de trabajo basado en interacciones que permiten tomar el “feedBack”, medir el progreso y re-ajustar prioridades en las fases de acuerdo a los cambios generados en el transcurso del proyecto. A continuación, se describen los Roles metodología Extreme Programming (XP):

- **Programador:** es aquel que escribe las pruebas unitarias y desarrolla el código fuente del proyecto.
- **Cliente:** Persona encarga de responder por las historias de usuario, e identifica cuales son de mayor prioridad de acuerdo con el modelo de negocio
- **Tester:** persona encarga a realizar las pruebas funcionales, y realizar la ejecución de las mismas
- **Tracker:** Proporciona realimentación (feedback) al equipo. También verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada una de las iteraciones.
- **Coach:** provee las buenas prácticas XP es el responsable de que todos los procesos se cumplan correctamente.
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.
- **Gestor:** Es el vínculo entre clientes y programadores. Ayuda a que el equipo trabaje de forma efectiva creando las condiciones adecuadas.

A continuación, se describen las principales fases de Extreme Programming:

- Fase de planeación.
- Fase de diseño.
- Fase desarrollo.
- Fase Ejecución de Pruebas.
- Fase Implementación.

3.2. Scrum

Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Retamosa. 2015).

Es una de las metodologías ágiles más utilizadas en la actualidad, utiliza un proceso iterativo que divide el desarrollo de un producto en ciclos de semanas llamados (Sprint). El equipo trabaja sobre una lista de requisitos priorizada, dando lugar, al acabar cada ciclo, a un producto entregable (George. 2016).

A continuación, se describe cómo podemos realizar la gestión de cambios desde Scrum, es importante resaltar que no debemos considerar cambios a las incidencias o bugs que aparecen a lo largo del proyecto, es por eso que en la metodología Scrum se definen tipos de cambios y los pasos a seguir para realizar la gestión de cambios de cada uno de estos.(Palacio, J 2015)

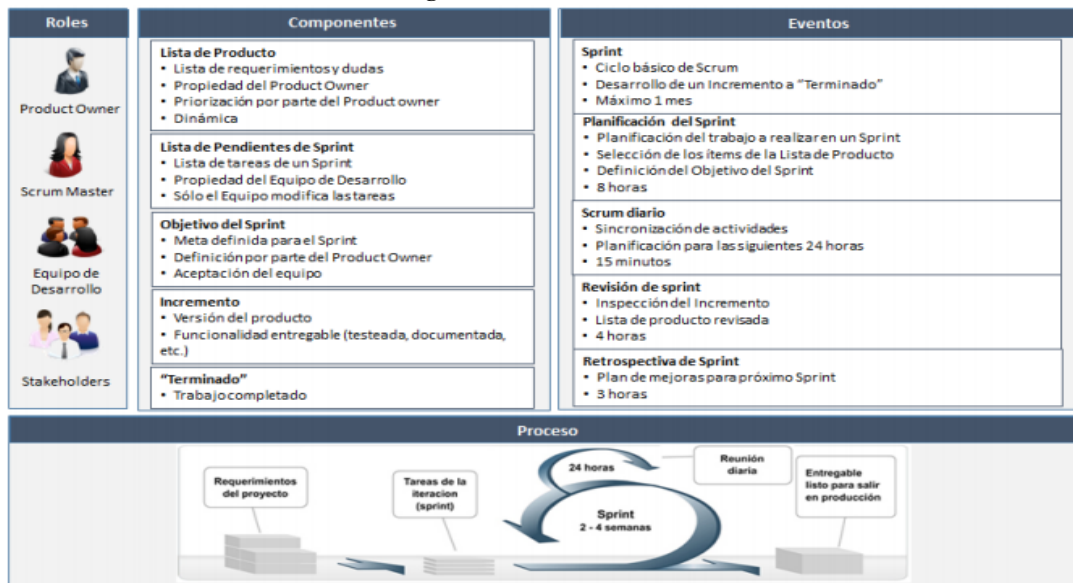
Scrum presenta dos tipos de cambios, los cambios normales son aquellos que tienen que ser revisados por el equipo de gestión de cambios y el Product Owner, estos cambios se pueden presentar a lo largo del proyecto pero no implican acción inmediata. Los cambios de emergencia son aquellos tan inesperados como los normales pero que requieren una acción en el mismo momento en el que llegan.

A continuación, se describen los pasos para adaptar la gestión de cambios en Scrum:

- Se formará el equipo del cambio, similar al Scrum Team, el cual será el encargado de preparar el Sprint Backlog con las acciones de gestión del cambio a realizar durante el primer Sprint.
- Se rellenará el Sprint Backlog con las acciones que el equipo considere que se deben tener en cuenta durante la realización del sprint que se esté desarrollando en ese momento.
- Antes, durante y después de la ejecución de los Sprint se deberán realizar las acciones que marca el framework: sprint backlog, planificación del sprint, daily y retrospectiva.
- En esta fase es vital la configuración del equipo de gestión del cambio, el cual debe estar formado por stakeholders vendedores y sabotadores que nos ayuden a ejecutar las acciones y que tengan la influencia necesaria para hacer llegar las acciones a los niveles necesarios.
- El equipo de gestión del cambio debe asegurarse que se implementan los cambios de forma incremental y de forma regular.
- De cada uno de los sprint se obtiene un aprendizaje continuo que sirve para mejorar y consolidar el cambio a medio y largo plazo.
- De forma transversal se llevarán a cabo las actividades de comunicación, creatividad, innovación, proceso participativo de decisión, etc. Estas actividades irán enriqueciendo los diferentes Sprint Backlog conforme vayamos avanzando en el proyecto.

A continuación, en la figura 3 se presentan los elementos que componen el marco de trabajo Scrum. Cada elemento dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso (Retamosa. 2015).

Figura 3. Proceso Scrum.



Fuente: Adaptado de (Retamosa. 2015).

3.3. Metodología de gestión de cambio CMOD integrado con EVM.

EVM (Earned Value Management) es una herramienta para realizar gestión de proyectos, su principal objetivo es medir el progreso en términos de alcance, costos y cronograma, midiendo el estado del cronograma actual con las actividades realizadas a la fecha, sin embargo, no cuenta con la gestión de cambio como parámetro significativo para efectuar la medición del progreso, sino en términos de cumplimientos de acuerdo a una planificación inicial, es decir que busca medir cuánto se alinea el progreso con el plan inicial, lo cual lo deja incompleto para realizar gestión de proyectos orientado al desarrollo de software. Por otro lado, se encuentra al CMOD, como una metodología que acepta la gestión de cambio como una parte vital y necesaria para los proyectos, representa el cambio como un retrabajo y evolución enfocado al alcance lo que permite una mejor visualización del progreso del proyecto de desarrollo de software.

Es importante conocer los procesos que establece CMOD con relación a la gestión de cambio y cómo detectan los problemas de costos serios, por ello, el artículo muestra cómo CMOD ayuda a aumentar la visibilidad del proyecto mediante la revelación de costos de retrabajo y evolución ocultos pero enormes. Con frecuencia en proyectos de software, el retrabajo y la evolución, incluyendo fallas y cambios, se registran en general en herramientas de seguimiento de problemas, pero su administración no está integrada en la gestión de proyectos. En el mejor de los casos, se tratan como problemas de calidad y solo se mantienen como métricas de calidad. Como tal, el enfoque cambia a la cantidad de problemas en lugar del esfuerzo dedicado a resolverlos. Por lo tanto, puede que no sea fácil darse cuenta de la cantidad de retraso y esfuerzo de evolución gastado en el proyecto. CMOD hace explícitos esos esfuerzos, los integra en la planificación y monitoreo del proyecto, ya que afectan significativamente.

CMOD es útil para realizar la gestión de cambio, ya que evidencia cómo se deben recopilar datos de esfuerzo y hacer que el seguimiento de problemas sea más sistemático, especialmente donde se encuentran más variedad de tareas por períodos de trabajo (Efe & Demirors. 2019).

3.4. Metodología de gestión de cambios de requisitos GSD.

Como ha sido mencionado anteriormente, es común que en un proyecto de desarrollo de Software esté sujeto a cambios debido diversos factores. La mala gestión de cambios es una de las principales causas de fracaso en estos proyectos (Shalinka. 2018) y sobre todo cuando se trata de desarrollo de software global. Richard Lai Neaveed Ali de la universidad La Trobe en Australia y Muhammad Azeem Akbar de la Universidad de Chongqing en China indican en sus artículos que la falta de comunicación, las diferencia horaria, los límites geográficos, las diferencias culturales y la falta de conciencia del trabajo realizado en diferentes sitios debido a la mala gestión del conocimiento son las causas de que la gestión de cambios

falle, por este motivo presento en su artículo una metodología para gestionar los cambios de requisitos para el desarrollo de software global (GSD) (Akbar; Shafiq; Ahmad; Mateen & Riaz. 2018) (Akbar; Sang; Khan; Mahmood; Qadri; Hu & Xiang. 2019).

La metodología de Richard Lai Neaveed Ali se basa en 3 etapas; compresión, análisis y finaliza con los cambios en los requisitos.

Para realizar la compresión de los requerimientos del proyecto, se obtiene la información sobre estos y se realiza un gráfico de los cambios teniendo en cuenta su tipo; agregación, eliminación o modificación. Este gráfico se estudiará con el fin de realizar la compresión por parte del equipo de desarrollo. En el análisis se examina qué módulos resultaron afectados, que sitios del GSD deberán realizar este cambio y que impacto tendrán las diferencias de estos como la cultura, la zona horaria, los recursos que necesiten cada sitio de trabajo, entre otros. Por último, se realiza la finalización del cambio donde se documentará el resultado de la gestión de cambios de requerimientos y se guardará en el repositorio central con acceso a todo el grupo de trabajo.

El autor aplicó esta metodología en un estudio de caso de un sistema de compras en línea, además también aplicó una gestión de cambio de requerimientos con el fin de comparar los dos resultados. El resultado que obtuvo Richard Lai Neaveed Ali es que su metodología disminuye los conflictos que ocurren durante la gestión de los cambios, disminuyendo el tiempo y el esfuerzo que se gasta para realizarlo. Además, el uso de los gráficos facilita el entendimiento de los requerimientos y sus cambios por parte de los diferentes sitios de trabajo del equipo de desarrollo permitiendo así un mejor entendimiento dentro del equipo (Ali & Lai 2016).

3.5. Metodología de gestión de cambios de requerimientos CRCM.

Otra de las metodologías que se encontró para realizar la gestión de cambios en los requerimientos es CMRC propuesta por Abeer AlSanad y Azeddine Chikh de la universidad de Riyadh en Arabia Saudita. Ellos expresan que para obtener éxito en la gestión de cambios de requerimientos se debe cumplir con 34 elementos detallados en el artículo, al reunir información de los artículos de las metodologías de CRM de la fecha se percataron que ninguno cumple con los 34 elementos especificados por tal motivo diseñaron y evaluaron CRCM.

CRCM es una metodología integrada de los modelos de RCM que se encontraron en los artículos, y cumple con los 34 elementos específicos. CRCM comienza diligenciando un formulario de solicitud de cambio que se registrara en la BD la cual tiene la referencia de los requisitos del proyecto, posteriormente se realiza un análisis de impacto para: costo, cronograma, recursos, funcionalidad, interesados y el esfuerzo que se debe realizar en el cambio. Más adelante, se verifica que la solicitud sea válida y se envía a la junta de control de cambios (CCB) donde se aceptará, suspenderá o se rechazará. Si se rechaza se notificará al solicitante, en caso de ser suspendida se le pedirá al solicitante más información detallada sobre el cambio y en caso de ser aceptada se registrará los requisitos dependientes y los entregables que resultan impactados por el cambio, se realiza el cambio, se verifica que el cambio esté correcto y se documenta.

En la evaluación que realizaron los autores se evidencio que a diferencia de las otras 3 metodologías con las que se comparó, CRCM cumple con los 34 elementos que se requieren para que CRM tenga éxito permitiendo que el cambio sea aplicado de una forma más rápida y sin presencia de tantos problemas (Abeer. 2017).

3.6. Metodología de gestión de cambios Kanban.

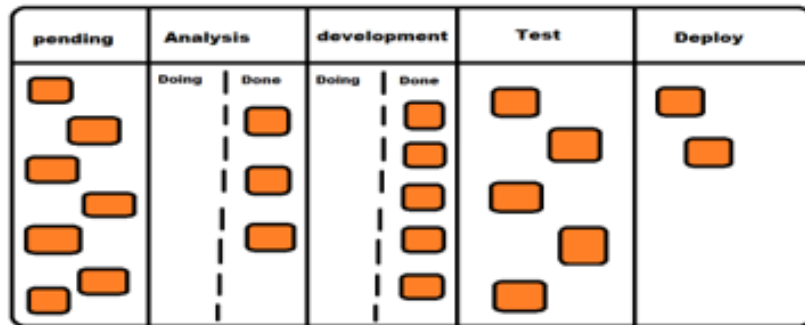
La metodología Kanban tiene un objetivo en realizar el trabajo correcto en el momento correcto, en este caso de proyectos de software los equipos de desarrollo de software tienen diferentes conjuntos de habilidades y velocidades de rendimiento en el trabajo. Los desarrolladores de proyectos comienzan implementando componentes de proyecto que agregan valor al proyecto de acuerdo con el estado de sus actividades planteadas (Lei, H.; Ganjeizadeh, F; Jayachandran, P. K., & Ozcan, P. 2017).

Kanban surge como alternativa para la gestión de cambios en proyectos de software incluyendo mecanismos que les permitan reaccionar de manera ágil a los continuos cambios que se presentan en sus entornos. Este método se adapta a las

necesidades de los proyectos rápidamente incrementando sustancialmente el nivel de productividad del equipo de proyectos debido a su enfoque evolutivo para gestionar las tareas de los cambios presentados. (Salvay, J. 2017)

Esta resulta ser combinada con la metodología Scrum el cual por medio de un ciclo de Sprint se listan las tareas de los cambios a realizar y así tener un seguimiento de las mismas. A continuación, en la Figura 5 se visualiza el tablero kanban con sus estados en las tareas planeadas enfocada en el proceso de gestión de cambios, que tiene un gran impacto en el flujo de trabajo debido a que permite tener una alta visibilidad del proyecto, al presentar de manera visual el estado de las tareas que los desarrolladores están llevando a cabo.

Figura 5. Tablero Kanban



Fuente: Autor Propio.

3.7. Metodología de gestión de cambios Scrumban.

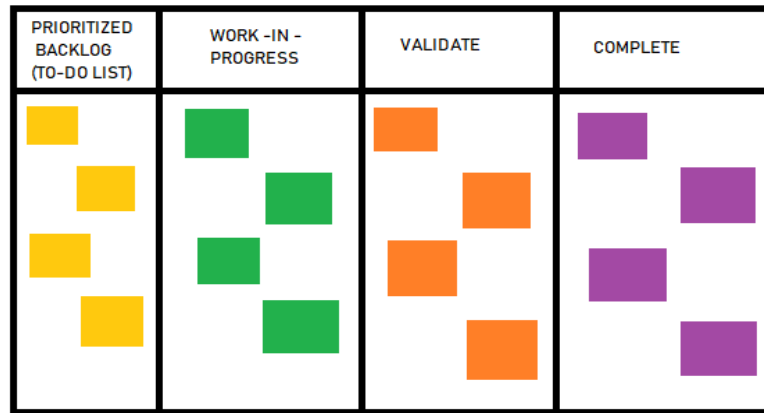
Esta metodología Scrumban intenta proporcionar un punto medio mezclando la estructura de Scrum y la planificación flexible de Kanban para crear una metodología ajustada a entornos en rápida evolución (García. 2015).

Una de las características de Scrum, es que las historias de usuario priorizadas y seleccionadas para ser trabajadas dentro de un sprint en particular, son un compromiso de entrega por parte del equipo hacia el cliente. Esto quiere decir que una vez que el sprint es iniciado, no son aceptados cambios en su contenido (es decir, en sus historias de usuario). En Scrumban, al ser un híbrido entre las dos metodologías esto no sucede de esta forma, ya que se utilizan las denominadas colas de trabajo de Kanban, que permiten que los sprints sean alterados cuando sea requerido, sin producir grandes impactos (Salvay, J. 2017). A continuación, se describen los principios básicos de Scrumban para el desarrollo de software en la gestión de cambios:

- Limitación de trabajo en proceso (WIP).
- Tirando valor a través del proceso de desarrollo.
- Hacer visible el proceso de desarrollo.
- Aumento del rendimiento.
- Usando una cartera de pedidos fija.
- Calidad de Incrustación.

A continuación, en la figura 6 se visualiza un ejemplo del tablero Scrumban en la cual se integra un sistema Scrum en un tablero Kanban con los estados de tareas en prioridad por hacer, trabajo en proceso o haciendo, validación o hecho, completadas o terminadas.

figura 6. Tablero Scrumban.



Fuente: Autor Propio.

Tabla 1. Cuadro comparativo metodologías con gestión de cambios en proyectos de software.

XP	SCRUM	CMOD	GSD	CRCM	KANBAN	SCRUMBAN
Se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define por proyectos con requisitos imprecisos, muy cambiantes y donde existe un alto riesgo técnico. Es por esto que crea prototipos para reducir riesgos.	Puede acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Su fuerte es la gestión de cambios con el producto y sprint backlog. Es entonces que se hacen evaluaciones de riesgos al inicio del proyecto y estos se revisan en las reuniones pertinentes de revisión.	Es útil para realizar la gestión de cambio, ya que evidencia cómo se deben recopilar datos de esfuerzo y hacer que el seguimiento de problemas sea más sistemático, especialmente donde se encuentran más variedad de tareas por períodos de trabajo.	Esta metodología disminuye el tiempo y el esfuerzo que se gasta para realizarlo. Además, el uso de los gráficos facilita el entendimiento de los requerimientos y sus cambios por parte de los diferentes sitios de trabajo del equipo de desarrollo permitiendo así un mejor entendimiento dentro del equipo	Cumple con los 34 elementos que se requieren para tener éxito, permitiendo que el cambio sea aplicado de una forma más rápida y sin tantos problemas. Sin embargo, evidenciamos que no cuenta con un mecanismo de comunicación ni de una rápida toma de decisiones, aunque sí cuenta con una alta flexibilidad de cambios.	Tiene un gran impacto en el flujo de trabajo ya que permite tener una alta visibilidad del proyecto. Sus principios en gestión de cambios consisten en reemplazar lo que está haciendo ahora con un proceso definido, prescrito a partir de un catálogo, diseñar un proceso de reemplazo adaptado a la dinámica de su entorno.	Modelo de desarrollo especialmente adecuado para proyectos de mantenimiento, proyectos en los que las historias de usuarios varían frecuentemente o en los que la complejidad o riesgo sea alto por poder surgir errores. Por ende, en su combinación de metodologías resulta más beneficioso adoptar el flujo de trabajo continuo propio de Kanban.

4. CONCLUSIONES

Para concluir, se tiene claro que las necesidades de los clientes cambian desde que se dan las indicaciones iniciales para el proyecto, por lo que se hace necesario percibir cuanto antes estos cambios. Por ende, estas necesidades tan cambiantes crean un nivel de incertidumbre, así que es lógico construir mecanismos y técnicas, para hacer partícipe en este proceso a los que más les interesa saber sobre el proyecto, es decir a los clientes. Con Scrum es posible compartir logros y la evolución del mismo.

Por todo esto, actualmente las metodologías ágiles del desarrollo de software toman como parte fundamental la idea de acoger el cambio como algo natural en los proyectos. Tenemos la metodología Scrum que es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar en equipo, y obtener el mejor resultado posible de un proyecto. No obstante, la metodología Kanban es complementaria a Scrum, en Kanban se visualiza el flujo de trabajo y este se tiene que dividir en tareas e incluirlas en el tablero. También se tienen las técnicas de XP que permiten flexibilidad y cambios en los requisitos. Finalmente, Scrumban es una metodología combinada y derivada de Scrum y Kanban, mezclando la estructura del Scrum y la planificación flexible de Kanban para crear entornos de rápida evolución. Por ende, analizamos que las metodologías ágiles pueden ser complementarias entre sí, de una u otra forma. Es decir, donde la metodología Scrum está pensada para aumentar la productividad, y XP para agregar valor y calidad al proyecto. Entre tanto, se infiere que todas se basan en el manifiesto ágil que establece sus principios metodológicos para desarrollar software, es allí donde se hace de gran importancia conocer estas metodologías que proponen ventajas enormes.

Entonces cómo se visualiza en este artículo, se han propuesto metodologías para la gestión de cambios en proyectos de software en los últimos 5 años, sin embargo, se detalla que para que estas tengan éxito se debe contar con una muy buena gestión de comunicación, una rápida toma de decisión y una flexibilidad para la modificación, la metodología que mejor cumplen con estos requerimientos es Scrum ya que al ser una metodología Ágil su objetivo es fomentar el trabajo en equipo, además de realizar reuniones periódicas y tareas interactivas para que el equipo esté al tanto y pueda adaptarse más fácilmente al cambio, asimismo de estar especialmente orientada para proyectos en entornos complejos, donde se necesita obtener resultados pronto y donde los requisitos son cambiantes o poco definidos. Su elemento principal para gestión de cambios es un Backlog de mejora organizativa (Backlog de Producto), donde el Backlog del Sprint estará integrado por aquellas mejoras (cambios) que se haya decidido implementar durante el próximo Sprint.

Inversamente se han encontrado dificultades para realizar la comparación de las metodologías con gestión de cambios en proyectos de software. Esto es debido a que no es trivial encontrar elementos comunes entre las metodologías expuestas y que además se halle información en donde se definan en completitud. Por ende, es complejo encontrar una base cuando se comparan metodologías de naturaleza diferente.

5. REFERENCIAS

- Abeer AlSanad, A. C. (2017). Software Requirements Change Management – A Comprehensive Model. En *Advances in Intelligent Systems and Computing* (Vol. 569). SpringerLink. doi: https://usc.elogim.com:2131/10.1007/978-3-319-56535-4_80.
- Ahmad, Z., Hussain, M., Rehman, A., Qamar, U., & Afzal, M. (2015, January). Impact minimization of requirements change in software project through requirements classification. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication* (pp. 1-5).
- Akbar, M. A., Sang, J., Khan, A. A., Mahmood, S., Qadri, S. F., Hu, H., & Xiang, H. (2019). Success factors influencing requirements change management process in global software development. *Journal of Computer Languages*, 51, 112-130.
- Akbar, M. A., Shafiq, M., Ahmad, J., Mateen, M., & Riaz, M. T. (2018, November). AZ-Model of software requirements change management in global software development. In *2018 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)* (pp. 1-6). IEEE.
- Ali, N., & Lai, R. (2016). A method of requirements change management for global software development. *Information and Software Technology*, 70, 49-67.
- Carranza, L. P. (2016). Gestión en Proyectos de Software. *Tecnología Investigación y Academia*, 4(2), 12-19.
- Chiapello, J., & Greiner, C. L. (2018). Gestión de cambios en proyectos software con evaluación de calidad de código fuente. In *XIX Simposio Argentino de Ingeniería de Software (ASSE)-JAIIO 47 (CABA, 2018)*.
- Dos Santos, A., Urdan, A. T., & Mangini, E. R. (2016). A gestão de mudanças como forma de melhoria da qualidade dos serviços de tecnologia da informação. *Revista Gestão & Tecnologia*, 16(1), 208-227.
- Efe, P., & Demirors, O. (2019). A change management model and its application in software development projects. *Computer Standards & Interfaces*, 66, 103353.
- García Rodríguez, M. J. (2015). Estudio comparativo entre las metodologías ágiles y las metodologías tradicionales para la gestión de proyectos software. Universidad de Oviedo. Obtenido de <http://hdl.handle.net/10651/32457>
- George Ellis, A. I. (2015). Chapter 8 - Agile Project Management: Scrum, eXtreme Programming, and Scrumban. En *Project Management in Product Development* (págs. 223-260). USA: ScienceDirect. doi:<https://doi.org/10.1016/B978-0-12-802322-8.00008-5>
- González Riscanevo, L. A. (2018). Prototipo para la gestión de cambios con base en el marco de trabajo COBIT 5. Bogotá: Universidad Distrital Francisco José de Caldas.
- Huércano, S. R. (2014). ITIL v3-Manual íntegro. línea]. Available: <http://www.biabile.es/wpcontent/uploads/2014/ManualITIL.pdf>. [Último acceso: 25 Febrero 2016].
- Kamal, T., Zhang, Q., & Akbar, M. A. (2019). Toward successful agile requirements change management process in global software development: a client–vendor analysis. *IET Software*.
- Lei, H., Ganjezadeh, F., Jayachandran, P. K., & Ozcan, P. (2017). A statistical analysis of the effects of Scrum and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, 43, 59-67.
- Microsoft. (2018). Change management. Obtenido de <https://www.microsoft.com/microsoft-365/partners/changemanagementframework>
- Mogogole, K. E., & Jokonya, O. (2018). A Conceptual Framework for Implementing IT Change Management in Public Sectors. *Procedia computer science*, 138, 835-842.
- Montero Fernández-Vivancos, G. (2016). Diseño de Indicadores para la gestión de proyectos. Valladolid: Universidad de Valladolid. Escuela de Ingenierías Industriales. doi:10.35376/10324/22086
- Motoa, G. (2015). Medición del éxito en los proyectos, una revisión de la literatura. *Ingenium*. Obtenido de <https://repository.usc.edu.co/handle/20.500.12421/776>

- Palacio, J (2015), Scrum Manager I v 2.5.1, Scrum Manager® (pp. 65 – 171)
- Project Management Institute, I. (2017). La guía de los fundamentos para la dirección de proyectos (Guía del PMBOK) / Project Management Institute. PMI.
- Retamosa Santos, A. (2015). Scrum. Aplicación del método ágil en la gestión de proyectos (Bachelor's thesis). Obtenido de https://e-archivo.uc3m.es/bitstream/handle/10016/26132/TFG_Ana_Retamosa_Santos.pdf?sequence=1&isAllowed=y
- Rivera González, A. A., & Dominguez Acevedo, O. A. (2019). Desarrollo de un aplicativo web, mediante metodología Scrum para la gestión y control de eventos relacionados en la universidad Santiago de Cali (Doctoral dissertation, Universidad Santiago de Cali).
- Shalinka Jayatilleke, R. L. (2018). A systematic review of requirements change management. En S. Jayatilleke, Information and Software Technology (Vol. 93, págs. 163-185). ELSEVIER. doi: <https://doi.org/10.1016/j.infsof.2017.09.004>
- Salvay, J. (2017). Kanban y Scrumban orientados a Proyectos de Tecnología de la Información. (pp. 43 – 123)
- Villa Morales, M. F (2019). ¿Es necesaria la gestión del cambio para la ejecución de los proyectos en la organización? (Ensayo presentado como requisito para optar al título de Especialista en Alta Gerencia, Universidad Militar Nueva Granada) (pp 10).