

UNA METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE EN UNA COMPAÑÍA FINANCIERA

Claudia Marcela Pérez Pérez

Cod. 1300926



Especialización en Gerencia Integral de Proyectos

Trabajo de Grado

Tutor
Angela Bacca

**Universidad Militar Nueva Granada
Bogotá DC / 2015**

UNA METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE EN UNA COMPAÑÍA FINANCIERA

A METHODOLOGY FOR AGILE SOFTWARE DEVELOPMENT IN A FINANCIAL COMPANY

Claudia Marcela Pérez Pérez
Ing. De Sistemas
Analista de desarrollo
Tecnom Colombia
Bogotá, Colombia
claumperez@gmail.com

RESUMEN

En este artículo se plantea una alternativa para mejorar el proceso de desarrollo de una compañía que actualmente no utiliza una metodología para la implementación de proyectos de software, lo que ha generado retrasos en las entregas, mala calidad de los productos terminados y sobrecarga laboral para algunos empleados. Las metodologías a plantear son metodologías de desarrollo existentes que ya han sido comprobadas como exitosas para la ejecución de este tipo de proyectos y que permiten dar cumplimiento a los requerimientos del cliente tanto en tiempo como en calidad aprovechando al máximo los recursos de la organización.

ABSTRACT

This article presents an alternative to improve the development process of a company that does not currently use a methodology for development software projects, which has led to delays in deliveries, poor quality of finished products and work overload for some employees. Methodologies to improve are development methodologies that have been proven as successful in development of this kind of projects and allow accomplishing customer requirements both in quality and time, maximizing the resources of the company.

INTRODUCCIÓN

En el desarrollo de software existen muchas falencias, una de ellas y la mas importante es la calidad con la que se entrega el producto al cliente y esto a causa de la dificultad para encontrar fallas en el desarrollo de los requerimientos (Merchán , Urrea, & Rebollar, 2008) y una de las soluciones a esto son las metodologías de desarrollo ágil ya que estas hacen énfasis en la entrega de valor al cliente y en la generación de negocio (Díaz, 2009)

La compañía a evaluar es una empresa que provee tecnología a entidades financieras, tiene varios proyectos de desarrollo de software, los cuales forman parte de la gerencia de soluciones informáticas, estos proyectos hacen parte de diferentes áreas que se dividen dependiendo del producto que se le ofrece a las entidades financieras, cada área está conformada por varios equipos que se encargan de los diferentes proyectos que están asignados, estos equipos tiene unos requerimientos o necesidades que provienen de un cliente externo (entidad financiera de un grupo financiero) u otra área; al momento de recibir estos requerimientos, se entregan al equipo y cada líder realiza una planificación (en la algunos de casos haciendo una reunión con su equipo) de tiempos de las diferentes etapas, el desarrollador se encarga de organizar su tiempo de acuerdo a los requerimientos que se le sean asignados, pero muchas veces por el afán de las entregas y las emergencias que se presentan por el camino, los tiempos tienden a no cumplirse, esto puede generar una entrega fuera del tiempo estipulado, o a tiempo pero en un estado de calidad no esperado por el cliente, además de generar una sobrecarga laboral para algunos empleados.

Analizando esta situación, se propone utilizar metodologías de desarrollo agiles, ya que una de las ventajas de esta estrategia está sobre el producto final el cual se podrá entregar en el tiempo y con la calidad esperada, utilizando los recursos de manera eficiente y generando mayor productividad, a su vez, se disminuye el riesgo de futuros defectos o errores que pueden aparecer por la falta de tiempo para realizar las pruebas correspondientes.

Uno de los factores que afectan los tiempos de entrega corresponde a los errores generados por desarrollos que ya han sido entregados, puesto que la mayoría del tiempo los desarrolladores están ocupados solucionando incidentes por falta de calidad en los mismos lo cual es consecuencia de la falta de pruebas en el desarrollo que se está haciendo ya que si existe un retraso en el tiempo planeado se opta por

disminuir el tiempo dedicado a las pruebas, generando así errores en el producto entregado al cliente (García, de Amescua, & Velasco, 2006).

Además con una implementación de una metodología de desarrollo se podrá generar una estructura en las áreas de trabajo donde sean menos rígidas y no solo se centren en un plan que se estipula al iniciar el proyecto, si no que sea flexible a los cambios que pueden ir presentándose, de acuerdo a las variaciones que puede pedir el cliente, y de esta forma el equipo sabrá cómo responder a estos cambios y el cliente quedará satisfecho generando así un lazo de confianza.

Actualmente en muchas compañías de software el proceso de desarrollo se realiza por medio de un tipo de metodología ya que por el avance tecnológico los clientes siempre esperan lo mas actual, lo mas nuevo, lo que implica la generación de un producto de una manera ágil pero que cumpla con todas las necesidades que el cliente espera de este. Esta investigación la considero muy útil, puesto que, al implementar estas metodologías en la compañía aumentará la productividad de los trabajadores y ayudará con el cumplimiento de los objetivos que tiene la empresa generando así confianza con los clientes y una mejor motivación en los empleados al ver que lo que ellos están desarrollando realmente es un producto de alta calidad. Siendo este uno de los mayores problemas en los proyectos en general, la desmotivación (Varas C, 2000).

Finalmente el impacto en la compañía será: mejoras en sus indicadores de productividad y reducción en los costos generados por los errores en los desarrollos, ya que estos no solo implican reprocesos innecesarios, sino que generan desconfianza en el sistema al cliente final. Adicionalmente estas metodologías están alineadas con los procesos de calidad que se está implementando ya que la documentación será realizada de la manera esperada dando mayor trazabilidad a la empresa (Ekas, 2012).

MARCO TEÓRICO

El concepto de metodología ágil aplicado al desarrollo de software nace después de una reunión que se realizó en febrero de 2001 en Utah-EEUU con 17 expertos de la industria de software, donde se discutió acerca de realizar un desarrollo de software rápido (Canós, Letelier, & Penadés, 2003) pero que a su vez cumpliera con las expectativas del cliente siendo flexible a los cambios que pueden generarse a través de la construcción del producto y que se centre en “Desarrollar software que funciona más que conseguir una buena documentación” (Canós, Letelier, & Penadés, 2003), siendo este uno de principios descritos en el manifiesto de la alianza creada por este grupo (The Agile Alliance, 2014).

Uno de los factores para que una empresa sea innovadora y competitiva en el mercado depende entre muchos factores, el poder de la negociación con el cliente como se expresa en la teoría de las 5 fuerzas de Porter (PORTER, 2008).

A través de la historia de la Ingeniería del software se pueden ver diferentes modelos que se han creado para la implementación de aplicaciones, entre estos podemos distinguir :

- En 1970 se planteó el *Modelo Cascada* donde se contemplan las etapas de: Ingeniería de sistemas, Análisis, Diseño, Codificación, Prueba, Utilización, Mantenimiento (Aparicio, 2012), uno de los problemas de esta metodología es que no existe una comunicación constante con el cliente, siempre se sigue de un paso al otro y si existe un error se notará en la etapa de pruebas, en este momento será necesario devolverse a la etapa inicial, lo cual tomará mas tiempo. Ya que en esta metodología luego de realizada una actividad no se repiten las tareas (véase Figura 1).

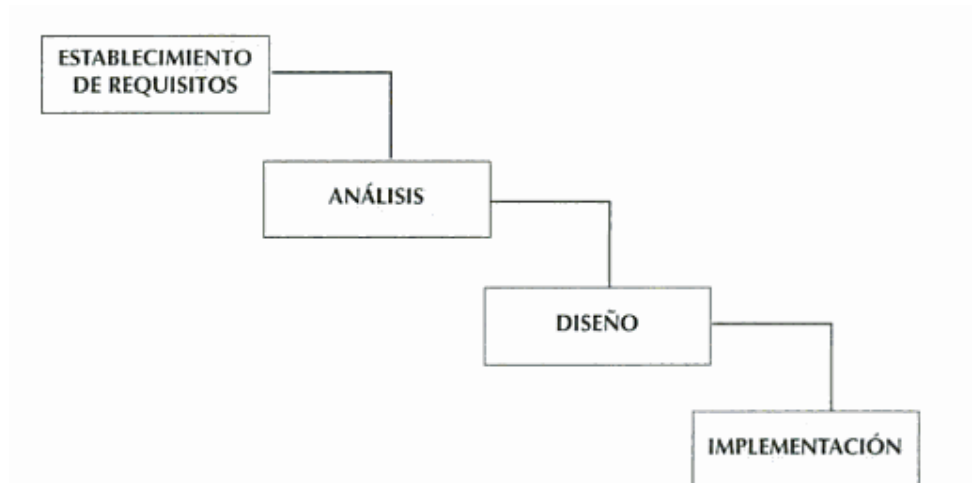


Figura 1. Modelo Cascada (Cortés Morales, 2006)

- En 1987 se planteó el *Modelo Espiral*, en este modelo se ve una nueva etapa que es el análisis de los riesgos y siempre se están haciendo ajustes al producto final (Aparicio, 2012), pero este modelo toma demasiado tiempo y tiende a entregar un producto en un tiempo mayor al que probablemente el cliente desee (véase Figura 2)..



Figura 2. Modelo Espiral (Cortés Morales, 2006)

- A través del tiempo se formuló otro modelo llamado *Extreme Programming* (XP), donde se plantea una metodología ágil, enfocada a la satisfacción del cliente y a reducir costos cuando existe un cambio ya que define 4 variables: costo, tiempo, calidad y alcance (Joskowicz, 2008), este modelo se orienta a una estrecha relación con el cliente en todo el proceso del desarrollo, lo cual

permite garantizar que el producto final cumpla con todos los requerimientos del cliente; sin embargo, puede generar altos costos en horas de desarrollo y algunos problemas para el trabajo en equipo ya que el cliente puede generar retrasos en el proceso.

Una vez teniendo claros los modelos de desarrollo, se dará paso a describir cuales son las metodologías de desarrollo ágil que empezaron a crearse, ya que conociendo el estado del arte con respecto al tema a investigar se podrá realizar un análisis de cual de estas metodologías es mas conveniente para implementar en un tipo de negocio financiero y a la estructura que se maneja en la compañía a estudiar.

Entre estas metodologías a tratar se encuentran:

- **Crystal Methodologies:** Creado por Alistair Cockburn en 1998 (Ramsin, 2009) . Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo (de ellas depende el éxito del proyecto) y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo Crystal Clear (3 a 8 miembros) y Crystal Orange (25 a 50 miembros) (Canós, Letelier, & Penadés, 2003).
- **Dynamic Systems Development Method6 (DSDM):** Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo el objetivo de crear una metodología RAD unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases (Canós, Letelier, & Penadés, 2003).
- **Adaptive Software Development (ASD):** Su impulsor es Jim Highsmith. Sus principales características son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la

tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo (Canós, Letelier, & Penadés, 2003).

- **SCRUM:** Scrum es una metodología ágil de desarrollo, un conjunto de practicas y reglas que abarca la transparencia y la adaptación de los requerimientos nuevos que nacen en medio de la construcción de otros, como suele suceder en muchas ocasiones donde el cliente tiene ideas nuevas con respecto a lo que solicitó (Schwaber, 2007).
- **RUP:** Metodología desarrollada y mantenida por Rational Corporation (compañía de IBM) "The Rational Unified Process® is a Software Engineering Process. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end-users, within a predictable schedule and budget. " (Rational Software).

En el caso de la metodología SCRUM, esta es una de las mas populares en muchas compañías ya que se tiene un mayor contacto con el cliente por las entregas tan frecuentes que se realizan, y esto hace que se tenga mayor adaptación a los cambios que puede pedir el cliente, además, se puede mitigar de una mejor manera los riesgos que se presenten en el transcurso del desarrollo del proyecto, y al analizar el tipo de negocio de la empresa a evaluar los requerimientos que usualmente se solicitan cambian constantemente y siempre hay poco tiempo para realizar estos cambios, lo cual es una ventaja que se tiene al implementar SCRUM, puesto que este está diseñado para proyectos donde se necesiten resultado rápidos y con requerimientos cambiantes (Rivadeneira, Vilanova, Miranda, & Cruz, 2013). Por esta razón se escoge esta metodología como parte del tema principal a investigar.

SCRUM

Es una manera ágil de administrar un proyecto de software (Mountain Goat), en donde el producto a crear se divide en pequeñas partes para poder ser construido en secuencia, y cada equipo se encarga de una parte para así centralizar su atención. Para poder trabajar en SCRUM es necesario tener 3 roles:

1. *Dueño del producto:* Quien determina que es lo que se va a implementar
2. *Equipo de desarrollo:* Quienes se van a encargar de desarrollar la necesidad del dueño del producto.

3. *Scrum Masters*: Lidera el proceso de implementación del producto para que este sea implementado de manera exitosa. (Scrum)

Antes de empezar a implementar esta metodología se debe tener en cuenta algunos aspectos, como lo es la cultura organizacional, ya que la idea es trabajar en equipo se debe tener una buena comunicación entre cada uno de los integrantes de los equipos, además, estos equipos deben ser pequeños entre 5 y 9 personas, las cuales deben ser estables en su trabajo (Proyectosagiles.org). Adicional a esto se debe contar con una buena relación con el cliente y con el o los proveedores, para facilitar la comunicación entre todos.

Scrum se compone de 3 fases, la primera es donde se realiza la planificación de lo que se va a hacer, donde se realiza la estimación de tiempo, se analiza el alcance y se realiza el diseño de la implementación a realizar, la segunda es el desarrollo del producto y la tercera es la preparación del producto final, donde se contemplan las pruebas para evaluar el correcto funcionamiento del producto desarrollado (Scrum Manager).

Para la etapa de planeación se realiza una reunión donde se determinan las necesidades del cliente y las prioridades de este, y se definen los requerimientos a trabajar (Barrios, V. Godoy, G. Fernández, I. Mariño, Ferreira, & Zarrabeitia, 2011).

La etapa de desarrollo del software se realiza en pequeñas fases llamadas "Sprint", esto representa el tiempo en el que se va a desarrollar una pequeña parte del producto final que el cliente quiere, este puede ser de un mes o menos y cuando termina un "Sprint" debe iniciarse el siguiente (Scrum Guides, 2013). Los requerimientos que se desarrollan en cada Sprint se definen en una reunión llamada "Daily Scrum" que se debe realizar cada día con el equipo, la duración de esta reunión debe ser de máximo 15 minutos y se realiza el seguimiento de las tareas que cada integrante del equipo está ejecutando. Al finalizar el Sprint se realiza otra reunión de máximo 4 horas donde se hace entrega de lo realizado en el Sprint al cliente por lo cual, entre los asistentes a esta reunión deben estar incluidos adicional al equipo, el cliente, el gerente y los usuarios. (Barrios, V. Godoy, G. Fernández, I. Mariño, Ferreira, & Zarrabeitia, 2011).

ARTEFACTOS

Son los productos de trabajo y en esta metodología se usan 3:

- *Product Backlog*: Documento de los requerimientos incluyendo estimación de tiempos (Ticona Yanqui, 2014)
- *Sprint Backlog*: Se describen las prioridades y la forma en que se va a implementar los requerimientos, y se definen las tareas (Ticona Yanqui, 2014).
- *Burndown Chart*: Es el grafico en el que se muestra el avance del Sprint (Ticona Yanqui, 2014).
- *Sprint Burn down*: Se muestra mediante un gráfico los requerimientos faltantes por realizar definidos en el Spring Backlog, esto da la posibilidad de informar la posibilidad de culminar el sprint con éxito (Tomaselli, Acuña, Estayno, & Lenkovich, 2013).
- *Sprint Burn up*: Grafica donde se muestra la evolución del desarrollo del producto (Villarreal, 2008).

Cuando se esta realizando el desarrollo en uno de los Sprint el Scrum Master se encarga de hacer el seguimiento al estado de cada una de las tareas a realizar, estas tienen 3 tipos de estado:

- *ToDo*: Hace referencia a una tarea que no se ha iniciado
- *InProgress*: Hace referencia a una tarea en la que un desarrollador se encuentra trabajando.
- *Done*: Hace referencia a una tarea finalizada (Nazareno, Leone, & Gonnet, 2013)

Al implementar esta metodología se puede obtener un incremento en la productividad ya se mejora el tiempo de respuesta a las peticiones de cambios hechas en el proceso de desarrollo (Villanueva, 2014).

METODOLOGÍAS A USAR EN LA INVESTIGACIÓN

En esta investigación se utilizará la metodología cualitativa dado que la investigación a realizar se basa en conceptos y la relación entre estos conceptos (KRAUSE, 1995), en este caso de la manera en que se debería realizar el proceso de desarrollo de software de con un tipo de estudio exploratorio puesto que el objetivo principal es descubrir (Grajales G, 2000) de qué manera se puede implementar una metodología en una compañía centrada en desarrollo por procesos y donde actualmente no existe una metodología de desarrollo de software, por tanto será necesario realizar una investigación para obtener datos sobre la situación actual con respecto al desarrollo de software y así poder formular una estrategia adecuada con los datos recolectados.

Esta investigación se realizará mediante una entrevista con algunos de los desarrolladores de las diferentes áreas de la compañía, para así saber la forma en que actualmente se está desarrollando, también una entrevista con algunos líderes y subdirectores de varios proyectos.

Para realizar las encuestas se hará un formato en la página de Monkey Survey (<https://es.surveymonkey.com/>) con el fin de facilitar la recolección de la información, este formato se enviará a 1 subdirector, 2 líderes y 3 desarrolladores de 3 proyectos de la compañía, se escogen estos roles ya que por un lado un subdirector es quien gestiona los proyectos de cada área y es quien tiene el conocimiento de las estadísticas actuales de los requerimientos entregados a los clientes, los líderes son quienes se encargan de estimar y planificar los requerimientos nuevos y gestionar el tiempo de su equipo, y los desarrolladores son quienes están en contacto directo con el proceso de elaboración del producto que se le entregará al cliente.

Se usará un método de muestreo no probabilístico por el límite de tiempo que se tiene.

Las preguntas que se van a realizar van a ser de tipo abiertas ya que se quiere saber el estado actual del proceso de desarrollo de software y que dificultades se presentan en este proceso, adicional, se quiere identificar el conocimiento que se tiene de la metodología planteada, esto con el fin de conocer las posibles dificultades que se pueden presentar al implementar la metodología.

Anexo 1. FORMATO ENCUESTA

Implementación de una metodología de desarrollo ágil

Nombres y

Apellidos: _____

Teléfono: _____

Cargo: _____

1. ¿Conoce alguna metodología de desarrollo ágil? ¿Cuál?
2. ¿Tiene experiencia con alguna metodología de desarrollo ágil?
Si respondió que si. ¿Cree usted que es viable implementar una metodología ágil en una empresa que está orientada a procesos?
3. Si tiene personas a cargo ¿Cuánto tiempo le dedica al seguimiento de las tareas o actividades de cada una de las personas a su cargo?
4. ¿Conoce la metodología Scrum?
5. En caso de conocer la Metodología Scrum ¿ Cree usted que es factible usar esta metodología en los proyectos de desarrollo de software de la compañía donde trabaja?
6. ¿Considera que actualmente se están cumpliendo los siguientes aspectos en las entregas al cliente?:
 - Tiempo
 - Calidad
 - Costo
 - Ninguna de las anteriores
 - Todas las anteriores
7. ¿Considera usted que al tener una mayor comunicación con el cliente a la que se tiene actualmente podría obstruir en el desarrollo de los requerimientos?
8. ¿Cree usted que si se dividen los proyectos en pequeñas tareas esto podría estabilizar la carga laboral de las personas del equipo?

ANÁLISIS

De acuerdo a los resultados obtenidos de las encuestas realizadas se puede analizar que:

- En la compañía se tiene conocimiento del tema de metodologías ágiles para desarrollar, esto será de gran ayuda en la implementación de nuevas metodologías.
- En cuanto a la viabilidad de la implementación de la metodología en una empresa orientada a procesos, la mayoría de personas encuestadas concordaron en que es viable y una propuesta interesante con respecto a esto es la de una metodología híbrida, por temas de estructura organizacional de la compañía donde se está realizando la investigación.
- Con respecto al seguimiento, Scrum es la metodología apropiada. Se obtuvieron resultados satisfactorios para la implementación de la metodología ya que según el cargo se está haciendo seguimiento constante, por lo cual las reuniones diarias propuestas por SCRUM (diarias), podrían realizarse sin alterar el tiempo actual que se dedica al seguimiento de tareas.
- En las entregas realizadas al cliente se puede observar que una de las prioridades es el tiempo, aspecto que actualmente se está cumpliendo, pero en cuanto a la calidad y los costos presupuestados esto puede estar variando en su cumplimiento, por esto se podría analizar que una metodología ágil podría ayudar a reforzar el aspecto de calidad del producto entregándolo en el tiempo en que se espera con menos costos.
- El contacto constante con el cliente, según los resultados, no parece ser un inconveniente y no obstruiría el trabajo de los desarrolladores, por el contrario esto podría ayudar a aclarar dudas que se tengan a tiempo para poder cumplir con las entregas.
- Según los resultados obtenidos se puede analizar que en cuanto a la división de los proyectos en tareas más pequeñas es una idea que al parecer es bien acogida por los diferentes roles, ya que sería de gran ayuda en los procesos de seguimiento y control.

- Según los resultados obtenidos también se puede observar que una solución más óptima, no solo sería la implementación de una metodología en específico, si no también se podría combinar con los aspectos que plantea otra metodología un poco mas rígida como lo es RUP.

Con esta información recolectada se puede determinar que estrategias serían las mas acordes para implementar esta solución, ya que por un lado tenemos un conocimiento previo del tema, y por otro se puede denotar que existe la disposición para enfrentarse a un cambio en la organización de la implementación de proyectos de software, puesto que uno de los grandes problemas que hay que enfrentarse cuando hay que entrar a cambiar una manera de las hacer las cosas es el miedo al cambio por que ya se maneja una cultura organizacional y no se conoce otra y esta es la zona cómoda para las personas y al cambiar esto se podrían recibir negativas por parte de las personas directamente afectadas.

Como primera parte de la estrategia que se debería usar, es la capacitación del tema de metodologías ágiles, y la especialización del tema de metodología Scrum y RUP. Luego de tener conocimiento teórico del tema, se debe pasar a una etapa de reestructuración, en donde se armen los equipos de desarrollo y se determinen los roles que se van a usar. Adicional a esto se debe determinar la arquitectura a usar, puesto que esta debe tener algunos cambios ya que debe acomodarse a el dinamismo que se va a manejar, al principio se haría un prototipo de arquitectura como lo plantea RUP (Carralero Colmenar, 2013).

Ya con una estructura base se empezaría a dividir los proyectos en partes pequeñas y seguir lo que propone Scrum pero en cada sprint se trabajarían determinados casos de uso (Tabares Bedoya, 2011) que serían los que se crean de los requerimientos pedidos por el cliente. Y se seguiría con los estándares propuestos por Scrum hasta finalizar el proyecto y entregar el producto final al cliente. Y al final de la fase de construcción se deben realizar pruebas de integración para comprobar la funcionalidad de las aplicaciones en conjunto (Pérez A., 2011).

En el caso de la comunicación con el cliente se seguirían los estándares que plantea Scrum y se mantendría contacto constante con el cliente para poder así reaccionar mas rápido a los cambios.

CONCLUSIONES

Luego de la investigación realizada se puede concluir que:

- Las metodologías ágiles de desarrollo es un tema que ya es conocido entre el negocio, por lo cual no se partiría de ceros al implementar alguna metodología.
- La metodología Scrum también es bastante popular en el medio por lo cual llegar a implementarla no sería una idea tan fuera de lugar en una empresa donde aún no existe una metodología estandarizada para todos los procesos.
- Teniendo en cuenta el tipo de negocio de la compañía y el resultado de las encuestas se puede concluir que para optimizar el desarrollo de las aplicaciones la mejor opción es implementar factores de dos metodologías: Scrum y RUP, ya que la primera genera dinamismo en los procesos y la segunda mejora la ruta crítica de los proyectos. Por esto se concluye que se debería realizar una combinación de los prácticas de dos metodologías, en este caso la mejor opción sería implementar una parte de Scrum (metodología dinámica) y otra parte la metodología RUP ya que esta es más rígida y tiene ventajas que podrían ser de gran ayuda, pues da la posibilidad de poder actuar frente a la ocurrencia de un acontecimiento que pueda retrasar el proyecto, pues con RUP se genera una lista con la ruta crítica del proyecto y se formulan estrategias para su manejo (Villanueva, 2014).
- La estandarización de una metodología de desarrollo para todos los proyectos de tecnología de la compañía puede generar una reducción de costos y mejoras en los servicios que se le ofrecen a los clientes.
- La comunicación frecuente con los clientes permite una definición clara del requerimiento lo cual conlleva a unas pruebas optimas para generar un producto final de acuerdo a los estándares propuestos por el cliente. Al implementar estas metodologías de desarrollo se garantiza que estas etapas de la programación se cumplan.
- La división de los proyectos en pequeñas tareas permite realizar un mejor control del proyecto lo permite un óptimo uso de los recursos de esta forma el trabajo realizado por los desarrolladores será mas eficiente.

Bibliografía

- Díaz, J. R. (Octubre de 2009). Las metodologías ágiles como garantía de calidad del software. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software* , 40-43.
- García, J., de Amescua, A., & Velasco, M. (2006). TOP 10 de factores que obstaculizan la mejora de los procesos de verificación y validación en organizaciones intensivas en software. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software* , 18-28.
- Merchán, L., Urrea, A., & Rebollar, R. (Enero de 2008). Definición de una metodología ágil de ingeniería de requerimientos para empresas emergentes de desarrollo de software del sur-occidente colombiano. *Revista Científica Guillermo de Ockham* , 37-50.
- Varas C, M. (2000). *Departamento de Ingeniería Informática y Ciencias de la Computación- Facultad de Ingeniería-Universidad de Concepción* . Obtenido de Gestión de Proyectos de Desarrollo de Software: <http://inf.udec.cl/~mvaras/gpis/apunteGPDS.pdf>
- Ekas, L. (2012). *5 beneficios de las metodologías ágiles en el desarrollo de sistemas de software*. Obtenido de https://www.ibm.com/developerworks/community/blogs/rationalspain/entry/5_beneficios_de_las_metodolog_C3_ADAs_C3_A1giles_en_el_desarrollo_de_sistemas_de_software_5?lang=en
- Canós, J., Letelier, P., & Penadés, M. (2003). *Métodologías Ágiles en el Desarrollo de Software*. Obtenido de http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf
- The Agile Alliance. (2014). *The Twelve Principles of Agile Software*. Obtenido de <http://www.agilealliance.org/the-alliance/the-agile-manifesto/the-twelve-principles-of-agile-software>
- PORTER, M. E. (2008). The five competitive forces that shape strategy. *Harvard business review* , 25-40.
- Aparicio, A. (2012). *UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA*. Obtenido de <http://datateca.unad.edu.co/contenidos/301404/301404.pdf>
- Joskowicz, J. (10 de 02 de 2008). *Instituto de Ingeniería Eléctrica*. Obtenido de Reglas y Prácticas en eXtreme Programming: <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Ramsin, R. (2009). *Sharif University of Technology*. Obtenido de Software Development Methodologies: http://sharif.edu/~ramsin/index_files/sdmlecture12.pdf
- Schwaber, K. (2007). Obtenido de http://www.volaroint.com/wp-content/uploads/dlm_uploads/2014/03/DC-VOLARO-Training-Scrum-What_Is_Scrum.pdf
- A Toda Hora S. A. (n.d.). From http://www.ath.com.co/portal/page?_pageid=573,4098614&_dad=portal&_schema=PORTAL
- Mountain Goat. (s.f.). *Scrum*. Obtenido de Mountain Goat software: <http://www.mountaingoatsoftware.com/agile/scrum>
- Scrum. (s.f.). *What is Scrum?* Obtenido de Scrum: <https://www.scrum.org/resources/what-is-scrum/>

Proyectosagiles.org. (s.f.). *proyectosagiles.org*. Obtenido de Requisitos para poder utilizar Scrum: <http://www.proyectosagiles.org/requisitos-de-scrum>

Rational Software. (s.f.). *Rational Unified Process*. Obtenido de Best Practices for Software Development Teams:
https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Scrum Manager. (s.f.). *Scrum Manager BoK*. Obtenido de Modelo original de Scrum para desarrollo de software:
http://www.scrummanager.net/bok/index.php?title=Modelo_original_de_Scrum_para_desarrollo_de_software

Barrios, W., V. Godoy, M., G. Fernández, M., I. Mariño, S., Ferreira, F. M., & Zarrabeitia, C. T. (Octubre de 2011). *Universidad Nacional de La Plata*. Obtenido de SCRUM: Experiencia de Aplicación en una Empresa de Desarrollo de Software del NEA:
http://sedici.unlp.edu.ar/bitstream/handle/10915/18745/Documento_completo.pdf?sequence=1

Ticona Yanqui, F. E. (2014). METODOLOGÍA SCRUM PARA EL DESARROLLO DE SOFTWARE Y GESTIÓN DE PROYECTOS EN LAS PEQUEÑAS Y MEDIANAS EMPRESAS DE LA CIUDAD DE JULIACA. *Revista científica Investigación Andina* .

Tomaselli, G. P., Acuña, C., Estayno, M., & Lenkovich, C. (2013). *CoNaIISI*. Obtenido de SCRUM: Una revisión de la literatura: <http://www.conaiisi.unsl.edu.ar/2013/199-502-1-DR.pdf>

Nazareno, R., Leone, H., & Gonnet, S. (2013). *CoNaIISI* . Obtenido de Un Modelo Conceptual para la Trazabilidad de Procesos Scrum:
<http://www.conaiisi.unsl.edu.ar/portugues/2013/163-530-1-DR.pdf>

Villanueva, J. S. (2014). SCRUM Y RUP: COMPARATIVA Y PROPUESTA METODOLÓGICA. *T.I.A Universidad Distrital Francisco Jose de Caldas* , 39-48.

Villarreal, G. L. (2008). Notas de Scrum. *LPMagazine* .

Scrum Guides. (2013). *The Scrum Guide*. Obtenido de The Sprint:
[http://www.scrumguides.org/scrum-guide.html#theoryfgfd.\(sdfgd\).gdfgfd.fgsdgdgfg:gdfg](http://www.scrumguides.org/scrum-guide.html#theoryfgfd.(sdfgd).gdfgfd.fgsdgdgfg:gdfg)

Cortés Morales, R. (2006). *Introducción Al Análisis de Sistemas Y la Ingeniería de Software*.

Grajales G, T. (2000 йил 27-Marzo). *Instituto Universitario Puebla*. From TIPOS DE INVESTIGACION:
http://www.iupuebla.com/Maestrias/M_E_GENERO/MA_Maestria_Genero/Jose_Miguel_Velez/Tipos%20de%20investigacion.pdf

KRAUSE, M. (1995). LA INVESTIGACIÓN CUALITATIVA: UN CAMPO DE POSIBILIDADES Y DESAFÍOS. *REVISTA TEMAS DE EDUCACION* .

Rivadeneira, S., Vilanova, G., Miranda, M., & Cruz, D. (2013). *Universidad Nacional de La Plata*. From El modelado de requerimientos en las metodologías ágiles:
http://sedici.unlp.edu.ar/bitstream/handle/10915/27196/Documento_completo.pdf?sequence=1

Rational Software. (s.f.). *Rational Unified Process*. Obtenido de Best Practices for Software Development Teams:

https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Carralero Colmenar, M. (2013). Comparativa entre dos metodologías de desarrollo software. *Revista Digital Sociedad de la Información* (43).

Tabares Bedoya, L. F. (2011). *Personalización de RUP para proyectos académicos de desarrollo de software*. Obtenido de Universidad EAFIT:

<https://repository.eafit.edu.co/handle/10784/2754#.VG1h61eG9Rl>

Pérez A., O. (2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM. *Revista inventum* .