

NECESIDAD DE METODOLOGÍAS HÍBRIDAS EN EL DESARROLLO DE SOFTWARE

M.Sc. Ing. Jhillma Portanda Zurita.
jhillma_p@hotmail.com

Universidad Nacional de “Siglo XX”
Carrera de Ingeniería Informática
Llallagua, Bolivia

Resumen- *El uso de una metodología en el desarrollo de software es uno de los factores más importante para alcanzar el éxito en un proyecto de desarrollo de software, muchas veces se llega a adoptar una metodología no permite cumplir a cabalidad con las exigencias del usuario puede ser que no se adapte a estas necesidades o que el equipo de desarrollo no la esté aplicando correctamente; por esta razón es que se plantea el diseño de una metodología híbrida que rescate las características más ventajosas de las metodologías tradicionales y las metodologías ágiles conocidas y permita de esta manera estructurar una metodología que se acomode a cualquier situación y permita hacerle frente hasta a los requerimientos más volátiles de los usuarios.*

Palabras clave - *Metodología, ágil, híbrido, metodología tradicional, software.*

Abstract - *The use of a methodology in software development is one of the most important factors to achieve success in a software development project, many times a methodology is adopted that does not allow to fully comply with the user's requirements. be that it does not adapt to these needs or that the development team is not applying it correctly; For this reason, the design of a hybrid methodology is proposed that rescues the most advantageous characteristics of traditional methodologies and known agile methodologies and thus allows structuring a methodology that adapts to any situation and allows to face even the requirements more volatile of users.*

Keywords - *Methodology, agile, hybrid, traditional methodology, software.*

INTRODUCCIÓN

Para asegurarse de que un proyecto de desarrollo de software tenga éxito; el elemento básico y fundamental lo representa la metodología de desarrollo, está provee de una dirección a seguir para la correcta aplicación de los demás elementos. Generalmente el proceso de desarrollo se asocia al estricto control del proceso tomando en cuenta la definición de roles, actividades y los resultados esperados de cada una de estas incluyendo un estructurado modelado del sistema y su respectiva documentación. Este esquema tan tradicional para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el

entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir de las buenas prácticas de la Ingeniería del Software, asumiendo el riesgo que ello conlleva. En este contexto, surgen como alternativa las metodologías ágiles como una posible respuesta para llenar ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños, estas aportan una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. Dentro de las metodologías más populares como son : RUP (Rational Unified Process, Proceso Unificado de Rational), XP

(eXtreme Programming, Programación Extrema) y Scrum. Se propone, extraer las mejores prácticas de cada una para brindar una alternativa que ayude a las empresas bolivianas; puesto que en nuestro país es cada vez más popular el uso de sistemas de información en todas las áreas de desarrollo y productividad.

MARCO CONCEPTUAL

En el área de Ingeniería de Software, el término *metodología* [1] se ha adecuado para referirse a un marco de trabajo empleado para estructurar, planificar y controlar el proceso de desarrollo de sistemas computacionales.

Por lo que al utilizar una metodología para desarrollo de software se espera que ésta pueda proveer un conjunto de prácticas y herramientas que faciliten el proceso de desarrollo; ofreciendo un producto que satisfaga las expectativas del cliente, con alta calidad y seguro.

METODOLOGÍAS TRADICIONALES

Algunos ejemplos de este tipo de metodologías son: Estructurado de Yourdon, OMT (*Object Modeling Technique*, Técnica de Modelado de Objetos), RUP, Métrica 3, entre otras. Las metodologías tradicionales contribuyeron positivamente al ser incrementales e iterativas; promovieron la asignación de roles dentro del equipo de desarrollo, facilitaron la división del sistema en varios subsistemas y fomentaron el reúso de componentes.

De manera general, las metodologías tradicionales consideraron la importancia de documentar el sistema, permitiendo así, entender, extender y mantener el software.

Sin embargo, tienen algunas desventajas, entre las cuales se puede señalar que se requiere de un alto grado de disciplina, no se tiene respuesta rápida a cambios, se genera documentación ampulosa e innecesaria en algunos casos y se invierte mucho tiempo en el modelado del sistema. Además estas metodologías tienen un plan de proyecto muy riguroso. Por lo que de manera general, no consideran que el análisis, el diseño y la construcción son impredecibles la mayoría de las

veces. (Hernandez, 2012)

METODOLOGÍAS ÁGILES

El esquema propuesto por las metodologías tradicionales no resulta ser el más adecuado para muchos de los proyectos actuales en donde el entorno del sistema es muy cambiante y se exige reducir drásticamente los tiempos de desarrollo. En este escenario nacieron las metodologías ágiles, que en esencia combinan una filosofía y un conjunto de directrices de desarrollo basados en el “Manifiesto Ágil”. La filosofía busca la satisfacción del cliente y la entrega temprana del software incremental; equipos de proyectos pequeños y con alta motivación; métodos informales y un mínimo de productos de trabajo.

Las directrices de desarrollo resaltan la entrega sobre el análisis y el diseño, la comunicación activa y continúa entre los desarrolladores y los clientes.

Las metodologías ágiles, al igual que las metodologías tradicionales poseen ciertas ventajas y desventajas.

Algunas de las ventajas de las metodologías ágiles es que presentan respuestas rápidas y efectivas al cambio. Tienen un plan de proyecto flexible y una gran simplicidad de manera general en el desarrollo.

Sin embargo, las metodologías ágiles generan poca documentación y no hacen uso de métodos formales. Algunas de las metodologías ágiles más conocidas son: *XP*, *Crystal* y *Scrum*. (Hernandez, 2012)

Para un mejor entendimiento de las diferencias entre las metodologías ágiles y tradicionales revise el cuadro comparativo: Metodologías tradiciones y ágiles (sección 9).

METODOLOGÍAS HÍBRIDAS

Hay una gran diversidad de metodologías, aunque todas ellas caen dentro de alguna de las dos clasificaciones mencionadas: ágiles o tradicionales.

Sin embargo, existe una nueva categoría: las metodologías híbridas.

Las metodologías híbridas pretenden retomar las

ventajas de las metodologías existentes, de tal forma que son una combinación de las mejores prácticas descritas en cada una de ellas.

Dentro las metodologías híbridas se puede mencionar a EssUP (*Essential Unified Process*, Proceso Esencial Unificado) (Ivar, 2007) como la pionera.

EssUP: es una metodología creada por Ivar Jacobson en el 2010, basada en el Proceso Unificado (PU) (Ivar J. , 2010) , los métodos ágiles y la madurez de procesos

EssUP es ágil porque no pretende imponer un proceso específico, además toma en cuenta que es necesario tener flexibilidad y respuestas rápidas ante los cambios.

Sin embargo, EssUP menciona que es necesario documentar y modelar en UML (UML Group, 2012) , con lo cual retoma una importante característica de las metodologías tradicionales.

Por lo que, EssUP es una metodología híbrida, aunque conceptualmente, ya que en la práctica el equipo de desarrollo de software que pretenda utilizar esta metodología debe seleccionar el modelo de ciclo de vida de desarrollo de software que mejor se adapte a sus necesidades, asignar los roles que crean convenientes y seleccionar las mejores prácticas; con lo cual se presenta un gran problema si no se tiene la experiencia y el conocimiento necesario para saber elegir las mejores prácticas dentro de la Ingeniería de Software y aplicarlas de manera adecuada en cada proyecto.

EssUP presenta una nueva tendencia en metodologías para desarrollo de software, ya que intenta retomar algunas ventajas de las metodologías tradicionales y de las ágiles, convirtiéndola en la primera metodología híbrida.

OBJETIVOS

La presente investigación tiene la finalidad de:

Demostrar la necesidad de fusionar las mejores prácticas de las metodologías tradicionales y ágiles en el desarrollo de software para obtener productos de calidad.

Para lo cual se requerirá cumplir con las siguientes tareas:

- Indagar acerca del uso de metodologías híbridas y sus resultados en otros países.
- Identificar las prácticas empleadas en el desarrollo de software de empresas dedicadas a este rubro en el eje troncal de Bolivia.
- Seleccionar las mejores prácticas aplicadas por las empresas de desarrollo del país identificando si pertenecen a las metodologías tradicionales o a las ágiles.

MÉTODOS

El trabajo se basa en realizar una investigación exploratoria tomando en cuenta que las investigaciones relacionadas al tema no corresponden a nuestro país; en primer lugar se ha llegado a establecer que una investigación parecida se llevó a cabo en México como parte de un programa de maestría por la maestrante: Eréndira M Jiménez-Hernández el año 2012 en donde además se plantean los pasos para una nueva metodología denominada META (Metodología tradicional - ágil) en donde se pudo observar las fases, etapas y roles en base a un análisis de las metodologías más usadas en México.

Respecto a Bolivia se conoce que la mayoría de las empresas utilizan metodologías ágiles; tal es el caso de JALASOFT que viene trabajando bajo el entorno de trabajo de SCRUM, lo cual demuestra que en su mayoría se dejó de lado el uso de metodologías tradicionales como el RUP que en su momento tuvo su auge por la robustez y completitud de sus prácticas, además las empresas actualmente deben acomodarse a plazos de entrega más cortos por lo que se trata de generar la menor cantidad de documentación posible subsanando esto con un proceso de pruebas más detallado y minucioso.

A partir de estos aspectos se puede afirmar que el desarrollo en un 90 % de los casos se realiza con metodologías ágiles; pero en base a experiencias

planteadas en un congreso de El Alto hace algunos años se dio a conocer que las metodologías que más aportan al resolver los problemas que se presentan al desarrollar Software en el orden de prioridad están:

En primer lugar RUP (que se considera una metodología tradicional)

Seguido de XP, en tercer lugar se encuentra SCRUM; por ultimo Kanban y otras.

Esto nos hace pensar que RUP no está obsoleto y si pudo resolver en su momento tantos problemas; porque no rescatar de esta metodología algunas prácticas que bien podrían servirnos en el futuro para llevar adelante proyectos de gran magnitud sobre todo. No todas las respuestas están en las metodologías ágiles pues como muchos las han adoptado y usado como si fueran la mejor opción también existen desarrolladores que las han cuestionado duramente; tal es el caso de Uno de los ataques más duros y tempranos contra las metodologías ágiles proviene de una carta de Steven Rakitin a la revista Computer, bajo la rúbrica “El Manifiesto genera cinismo”.

Refiriéndose a la estructura positiva del Manifiesto, Rakitin expresa que en su experiencia, los elementos de la derecha (vinculados a la mentalidad planificadora) son esenciales, mientras que los de la izquierda sólo sirven como excusas para que los hackers escupan código irresponsablemente sin ningún cuidado por la disciplina de ingeniería. Así como hay una lectura literal y una estrecha del canon de CMM, Rakitin practica una “interpretación hacker” de propuestas de valor tales como “responder al cambio en vez de seguir un plan” y encuentra que es un generador de caos. La interpretación hacker de ese mandamiento sería algo así como: “¡Genial! Ahora tenemos una razón para evitar la planificación y codificar como nos dé la gana”.

• RESULTADOS

A partir de las averiguaciones realizadas se ha llegado a abordar a los siguientes resultados:

- Se cuenta con un antecedente en México para el planteamiento de metodologías híbridas de desarrollo de software, lo cual hace suponer que esta preocupación también se da en otros contextos.

- Se conoce que a nivel nacional la mayoría de las empresas utilizan metodologías ágiles y dejaron de lado a las metodologías tradicionales sin tomar en cuenta que se podrían rescatar algunos aspectos de las mismas en pro de mejorar la calidad de los productos desarrollados.
- Las metodologías ágiles no cuentan todavía con un entorno de trabajo bien definido incluso algunas de las afirmaciones del manifiesto ágil pueden llegar a causar incoherencia en su aplicación.

TABLAS Y FIGURAS

Cuadro comparativo: Metodologías tradicionales y ágiles:

Metodologías Tradicionales	Metodologías Ágiles
- Rigidez ante los cambios, de manera lenta o moderada - Los clientes interactúan con el equipo de desarrollo mediante reuniones - Grupos de gran tamaño y varias veces distribuidos en diferentes sitios - Dependencia de la arquitectura de software mediante modelos - Poco Feedback lo que extiende el tiempo de entrega - Mínimos roles - Basadas en normas de estándares de desarrollo - Procesos muy controlados por políticas y normas - Seguimiento estricto del plan inicial de desarrollo	- Flexibilidad ante los cambios del proyecto de forma moderada a rápida - Los clientes hacen parte del equipo de desarrollo - Grupos pequeños (promedio 10 participantes in situ) en el mismo lugar. - Menor dependencia de la arquitectura de software - Continuo Feedback acortando el tiempo de entrega - Diversidad de roles - Basadas en heurísticas a partir de prácticas de producción de código - Procesos menos controlados, pocas políticas y normas - Capacidad de respuesta ante los cambios

Cuadro 1: Cuadro comparativo

Fuente: <http://metodologiasagiles.wikispaces.com>

DISCUSIÓN

Tener metodologías diferentes para aplicar de acuerdo con el proyecto que se desarrolle resulta una idea interesante. Estas metodologías pueden involucrar prácticas tanto de metodologías ágiles como de metodologías tradicionales. De esta manera podríamos tener una metodología para cada proyecto, la problemática sería definir cada una de las prácticas, y en el momento preciso definir parámetros para saber cuál usar.

Es importante tener en cuenta que el uso de un método ágil no es para todos. Sin embargo, una de las principales ventajas de los métodos ágiles es su peso inicialmente ligero y por eso las personas que no estén acostumbradas a seguir procesos encuentran estas metodologías bastante sencillas y por lo tanto agradables.

CONCLUSIONES

A partir de esta pequeña investigación se llega a concluir que:

- No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software.
- Las metodologías tradicionales históricamente han intentado abordar la mayor cantidad de situaciones del contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños y de requisitos cambiantes.
- Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos.
- Las metodologías ágiles se caracterizan por su sencillez, tanto en su aprendizaje como en su aplicación; sin embargo, gozan tanto de ventajas como de inconvenientes.
- A pesar de las continuas críticas que las metodologías ágiles sufren, son usadas por muchas grandes empresas y se han utilizado en grandes sistemas, por lo que enriquecerlas

con las prácticas más estables de las metodologías tradicionales las enriquecerían y fortalecerían más.

Bibliografía

Hernandez, E. J. (01 de enero de 2012). *Revista Unam.MX*. Recuperado el 5 de julio de 2015, de Revista Digital Universitaria: Ivar, J. (12 de marzo de 2007). *The essential Unified Process*. Recuperado el 05 de julio de 2015, de http://www.ivarjacobson.com/process_improvement_technology/essential_unified_process_software/

Ivar, J. (09 de septiembre de 2010). *The unified Process Lifecycle*. Recuperado el 05 de Julio de 2015, de http://www.ivarjacobson.com/Unified_Process_Lifecycle.aspx

UML Group. (15 de octubre de 2012). Recuperado el 05 de julio de 2015, de <http://www.uml.org> <http://www.revista.unam.mx/>