

Recibido: 21 de septiembre de 2015
Aceptado: 30 de octubre de 2015**MODELO DE DESARROLLO ÁGIL DE UN “CLOUD ERP” PARA LAS
PEQUEÑAS Y MEDIANAS EMPRESAS (PYMES) DE NORTE DE SANTANDER****AGILE DEVELOPMENT MODEL OF A “CLOUD ERP” FOR SMALL AND
MEDIUM SIZED BUSINESS (SMB) OF NORTE DE SANTANDER****MSc. Fredy H. Vera R., MSc. Boris R. Perez G.****Universidad Francisco de Paula Santander, Grupo de Investigación GIDIS.**

Ave. Gran Colombia, No. 12E-96, Barrio Colsag. San José Cúcuta.

Tel.: (+577) 577 6655 Ext. 201-203.

E-mail: {fredyhumbertoverta, borisperezg}@ufps.edu.co

Resumen: En este artículo se propone un modelo que establece las fases, métodos, procedimientos y herramientas necesarias para estructurar y controlar el desarrollo de un ERP en la nube para las PYMES (Pequeñas y medianas empresas) de Norte de Santander. Este trabajo se realizó en 4 fases: primero se realizó una fundamentación teórica, luego una revisión de las metodologías de desarrollo ágil y por último se plantea el modelo de desarrollo ágil propuesto para la creación del ERP para las PYMES.

Palabras clave: Cloud ERP, computación en la nube, desarrollo ágil, desarrollo de software basado en componentes.

Abstract: This paper deals with establishes a model of the phases, methods, procedures and tools necessary to structure and control the development of a cloud ERP for SMEs (Small and Medium Enterprises) of Norte de Santander is proposed. This work was performed in 3 phases: first a theoretical foundation was made, after a review of agile development methodologies and finally the agile development model proposed for the creation of ERP for SMEs arises.

Keywords: Cloud ERP, cloud computing, agile development, component-based software.

1. INTRODUCCIÓN

Los sistemas administrativos son herramientas vitales en el funcionamiento de cualquier organización, se utilizan para procesar las operaciones de la empresa y para realizar análisis posteriores con la información capturada, con el fin de tomar decisiones y establecer mecanismos para mejorar la competitividad.

Se estima que la inversión en Latinoamérica en aplicaciones ERP en el 2014 está cerca de los 1.931 millones de dólares (Presario TI Latinoamérica, 2011) y cada año viene en aumento.

El acceso por parte de las pequeñas y medianas empresas (PYMES) a los ERP es muy reducido en nuestra región. Para adquirir e implantar un ERP se requiere una alta inversión, se requiere comprar equipos de cómputo especializado, comprar las licencias para el funcionamiento, se incurre en costos de la puesta en marcha del sistema, se requiere además la contratación de personal calificado para soportar la infraestructura tecnológica y al sistema implantado, también se debe adquirir programas de capacitación para el personal de la empresa que va a usar el sistema. Con esta propuesta las PYMES se benefician porque pueden contar con una alternativa local o regional para la adquisición de un sistema ERP y

poder acceder a los beneficios que les ofrece a un costo bajo en comparación a otras alternativas.

Las principales soluciones que se encuentran en el mercado de los sistemas ERP son: Oracle E-Business Suite, Oracle JD Edwards Enterprise One., Adempiere, SAP, Microsoft Dynamics, Epicor y NetSuite, siendo algunos software libre como Epicor y Adempiere. Los costos de implementación y puesta en marcha de cualquier sistema ERP son altos, pero los beneficios que trae a la empresa son significativos y se puede obtener el retorno de la inversión en un corto tiempo.

Ahora el desarrollo de un sistema ERP es un proceso complejo, En varios proyectos de este tipo se evidencia la falta de sincronización entre el cliente y el grupo de desarrollo, causando demoras en la implementación, el sistema o módulo no se entrega a tiempo, se gastan más recursos y los clientes quedan insatisfechos. En la figura 1 se puede apreciar mejor esta problemática. Se presenta complejidad en los lenguajes, en las herramientas utilizadas para el desarrollo y en el mismo proceso de desarrollo.

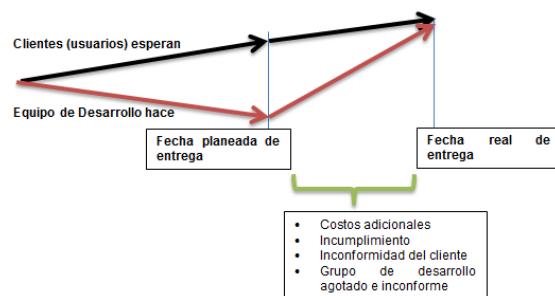


Fig. 1. Problemática en el desarrollo de sistemas ERP.

En respuesta a estos problemas y a la complejidad que se presenta, surge entre otras alternativas, las metodologías de desarrollo ágiles y el desarrollo basado en componentes, las cuales son base para los planteamientos hechos en este trabajo de investigación, donde se propone un modelo de desarrollo ágil para la implementación de un *cloud* ERP para las PYMES de Norte de Santander.

Se pretende usar la computación en la nube para reducir los costos de implantación, uso y de la infraestructura tecnológica que da soporte al sistema ERP en la nube, un "cloud ERP", el cual se puede definir como un sistema de información distribuido como software como servicio que garantiza una alta disponibilidad, que gestiona sus recursos de computo de una manera automática que realiza un balanceo de carga y se adapta a las

necesidades de cómputo y almacenamiento que requiere el cliente que lo utiliza, este cliente no necesita instalar nada local e ingresa a través de internet para usar sus servicios. En conclusión es un sistema ERP que usa los beneficios de la computación en la nube para un mejor y óptimo funcionamiento.

2. METODOLOGÍA

El presente trabajo de investigación se basa principalmente en los conceptos de la Investigación descriptiva y la investigación tecnológica aplicada.

Para el caso de la presente investigación se pretende describir y analizar las metodologías ágiles para el desarrollo de software, estudiando los métodos que existen en la actualidad, también se analiza la aplicación de estas metodologías en el desarrollo de aplicaciones para la Nube y en el desarrollo de sistemas ERP en nuestra región, luego se propone el modelo de desarrollo. En la figura 2 se puede ver el modelo llevado a cabo para realizar esta investigación.

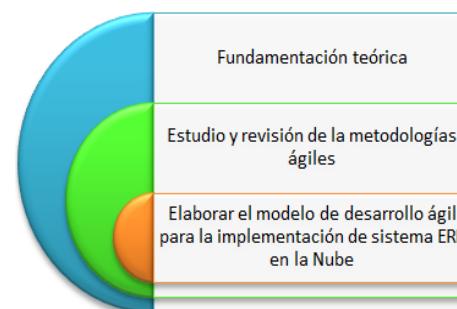


Fig. 2. Modelo de la investigación

Se inicia la investigación con recopilación de información necesaria para tener los fundamentos teóricos, después se realiza un diagnóstico del problema que se pretende resolver, luego se realiza el estudio y análisis de las metodologías ágiles existentes para poder por último elaborar el modelo de desarrollo ágil para la implementación de sistemas ERP en la nube.

3. RESULTADOS

El resultado principal de este trabajo de investigación corresponde al modelo de desarrollo ágil propuesto para implementar en un mediano o corto plazo un Cloud ERP para las PYMES del Norte de Santander.

3.1 Revisión y análisis de las metodologías ágiles

Este análisis y revisión se realizó en dos etapas, la primera consistió evaluar el nivel de uso por parte de los desarrolladores de software de Santander y Norte de Santander de las metodologías ágiles y en la implementación de sistemas ERP. La segunda etapa consistió en realizar la revisión bibliográfica y comparación de las metodologías ágiles.

3.1.1. Nivel de uso por parte de los desarrolladores de software de Santander y Norte de Santander de las metodologías ágiles y en la implementación de sistemas ERP

Se entrevistaron a 69 desarrolladores senior de software y se obtuvieron los siguientes resultados.

El 61% de los encuestados conocen las metodologías ágiles, pero solo el 26% las han usado en los proyectos de desarrollo, las metodologías ágiles más usada son Scrum y XP, siguiendo la tendencia nacional y mundial. El 61% manifiestan tener un nivel de conocimiento entre medio y alto sobre las metodologías ágiles.

Para cerca del 77% el número de proyectos en los cuáles han usado metodologías ágiles es menor a 5, siendo este un nivel bajo de uso de las metodologías de desarrollo ágil.

En cuanto a los sistemas ERP, el 55% de los ingenieros conoce los sistemas ERP y la gran mayoría (87%) de los ingenieros que los conocen manifiestan tener un nivel de conocimiento entre medio y bajo sobre estos sistema, sólo el 10% tienen un nivel de conocimiento alto. Sólo el 22% de ingenieros encuestados han participado en proyectos de desarrollo de ERP y tan solo el 6% han usado las metodologías ágiles en su desarrollo, otro 6% ha usado una combinación de las metodologías ágiles y tradicionales.

En cuanto a la implantación o implementación de un ERP comercial al indagar si habían participado en un proyecto de este tipo sólo el 19% de los encuestados ha participado.

Este porcentaje indica que el uso de sistemas ERP en la región es bajo, por lo tanto se puede formular la pregunta *¿Qué tecnologías o sistemas están usando las empresas de la región? ¿Es necesario el desarrollo de un sistema ERP regional que mejore la gestión administrativa de las empresas?*

3.1.2. Descripción y comparación de las metodologías de desarrollo ágil

Las metodologías ágiles son métodos de ingeniería del software basados en el desarrollo iterativo e incremental. Los requerimientos y soluciones evolucionan mediante la colaboración de grupos autoorganizados y multidisciplinarios. Se centran en la minimización de los riesgos y control del cambio. Se desarrolla en iteraciones, en cada iteración se obtiene un prototipo funcional del sistema (Cunningham, Medinilla, Giné, & Gómez, 2001).

Las metodologías ágiles de desarrollo estudiadas son: XP – Programación extrema, Scrum, ASD – *Adaptative Software Development* y las metodologías Cristal.

XP fue formulada por Kent Beck (Beck, 1999). Define un proceso iterativo e incremental con pruebas unitarias continuas y entregas frecuentes. El cliente o un representante del cliente son integrados al equipo de desarrollo (Jeffries, 2014).

Scrum es un proceso donde se aplican un conjunto de buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto. Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto (Proyectos agiles ORG, 2014), (Schwaber & Beedle, 2001).

Las metodologías Cristal fueron desarrolladas por Alistair Cockburn. Es un conjunto de metodologías caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. (Highsmith, 2002) (crystalmethodologies.org, 2013)

ASD - *Adaptative Software Development* su impulsor fue Jim Highsmith. Es iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tres fases esenciales: especulación, colaboración y aprendizaje. (Highsmith, 2002)

Ahora se presenta un resumen de comparar los siguientes factores, basados en (Highsmith, 2002), (Riehle, 2000), (Pérez Pérez & Gonzalez Cabrera).

1: Enfoque de desarrollo: XP y Scrum son iterativos e incrementales, ASD es iterativo y Cristal es incremental.

2: Tiempo de iteración recomendado: para XP es entre 1 a 6 semanas, para Scrum es de 2 a 4 semanas, para ASD es de 4 a 8 semanas y para Cristal depende de la familia que se escoja, la más corta corresponde a un par de semanas.

3: Equipo de desarrollo: XP se recomienda para equipos pequeños, menos de 20 miembros, Scrum para cualquier tamaño, ASD para equipos pequeños de 5 a 9 miembros y Cristal se adapta a cualquier número de miembros, según el número de miembros se escoge la familia cristal a usar.

4: Tamaño del proyecto: XP y ASD funcionan mejor con proyectos pequeños, Scrum se adapta a cualquier tamaño de proyecto al igual que Cristal.

5: Documentación: Las 4 metodologías solo exigen una documentación básica, la necesaria para describir el proyecto.

6: Ventajas: XP: El cliente hace parte del equipo de desarrollo, buenas prácticas bien definidas, hay realimentación en cada iteración. Scrum: Alto nivel de colaboración y comunicación, realimentación continua con el cliente. ASD: Se desarrollan primero los componentes de mayor riesgo, posee un ciclo de aprendimiento donde se analiza lo sucedido en cada iteración. Cristal: Tiene una familia de metodologías que se adaptan a cualquier tipo de proyecto y tamaño.

7: Desventajas: XP, Scrum y ASD tienen una débil documentación. En XP se puede presentar una ausencia de disciplina y la presencia del cliente es obligatoria. Scrum no tiene un control sobre el proyecto adecuado. En cristal una ineficiencia en la coordinación de equipos grandes.

3.2. Modelo de desarrollo ágil para la implementación del Cloud ERP

Teniendo en cuenta los principios del desarrollo ágil descritos en la sección anterior, y el desarrollo basado en componentes, se plantea el siguiente modelo para desarrollar y mantener un ERP para las PYMES distribuido en la nube. El modelo permite una mayor agilidad en el desarrollo y flexibilidad para que las PYMES armen su ERP. El modelo de desarrollo propuesto se puede apreciar en la figura 3.

El desarrollo basado en componentes es un enfoque que permite el desarrollo de software mediante la existencia de componentes de software preexistentes. Este enfoque conlleva grandes

cambios en la forma de desarrollar (CrnKovic, Chaudron, & Larson, 2006). Entre las ventajas del modelo de desarrollo basado en componentes se encuentran la mejora en la producción de software y la disminución del costo de desarrollo, además de disminuir el tiempo de mantenimiento y pruebas (Geisterfer & Ghosh, 2006).

A continuación se detallan las fases del modelo propuesto.

Planificación: En esta fase se deben describir y detallar las necesidades del cliente, esta descripción es preliminar y puede cambiar a medida que avance el proyecto, adaptándose al final de cada iteración en la fase de construcción. El resultado de esta fase es un documento donde se establezcan los módulos a desarrollar y sus prioridades definidos en compañía del cliente.

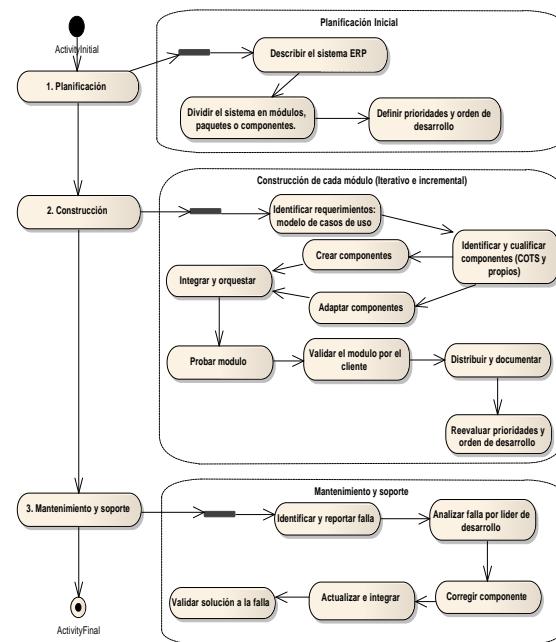


Fig. 3. Modelo de desarrollo del cloud ERP.

Construcción: Esta fase es la más importante y la que requiere mayor trabajo, se realiza de forma iterativa e incremental, cada módulo desarrollado debe ser probado por el cliente y puesto en producción inmediatamente. La construcción se realiza basada en componentes. El repositorio de componentes es la pieza principal del desarrollo basado en componentes, ya que permite la administración, selección, identificación, cualificación y ensamblaje de componentes de software (Rojas Morales, Delgado Rojas, & Vera Rivera, 2011). Para crear componentes software se puede usar el proceso de desarrollo propuesto por

(Vera Rivera & Rojas Morales, 2010). Al finalizar el desarrollo e integración de los componentes se planea la próxima iteración, evaluando las prioridades y necesidades del cliente, se puede cambiar el orden de desarrollo propuesto inicialmente.

Mantenimiento: Esta fase permite la evolución del sistema a medida que es utilizado por el cliente, se van identificando fallas, mejoras y cambios en el funcionamiento del sistema. A medida que se van realizando las iteraciones, van quedando módulos en producción los cuales son utilizados por el cliente y los usuarios finales. Este proceso debe ser automatizado, existen herramientas software como Bugzilla que permiten llevar a cabo este proceso (bugzilla.org, 2014).

CONCLUSIONES

Los beneficios que trae la implementación de un Cloud ERP para las PYMES de la región sería considerable, los costos de implementación, soporte y mantenimiento al ser una solución en la nube se reducen considerablemente.

El nivel de uso de las metodologías agiles en la región es bajo ya que se utilizan en pocos proyectos, la mayoría de ingenieros aunque conoce de metodologías ágiles no las usa en sus proyectos. La gran mayoría manifiestas que las metodologías ágiles mejoran el tiempo de desarrollo y la calidad del producto final, siendo la falta de experiencia en el uso y la falta de disciplina para seguir la metodología los principales inconvenientes en su adopción.

Antes de iniciar la implementación del Cloud ERP es importante analizar la viabilidad y validar si en realidad las PYMES de la región necesitan y están dispuestas a adquirir un sistema ERP local o regional.

Se recomienda formular un proyecto que resuelva los siguientes interrogantes *¿Qué tecnologías o sistemas están usando las empresas de la región? ¿Es necesario el desarrollo de un sistema ERP regional que mejore la gestión administrativa de las empresas?*

El modelo de desarrollo propuesto debe validarse y probarse para garantizar su correcto funcionamiento y depurarlo de posibles fallas.

REFERENCIAS

- Beck, K. (1999). *Extreme Programming Explained. Embrace Change*. Pearson Education.
- bugzilla.org. (2014). bugzilla.org. Recuperado el 19 de 8 de 2014, de <http://www.bugzilla.org/>
- CrnKovic, I., Chaudron, M., & Larson, S. (2006). Component - Based Development Process and Component Lifecycle.
- crystalmethodologies.org. (2013). Recuperado el 25/4/14: <http://www.crystalmethodologies.org/>
- Cunningham, W., Medinilla, A., Giné, A., & Gómez, E. (2001). Manifesto for Agile Software Development. Recuperado el 14 de agosto de 2010, de <http://agilemanifesto.org/>
- Geisterfer, C., & Ghosh, S. (2006). Software component specification: A study in perspective of component selection and reuse. Proceedings of the fifth international conference on comercia-off-the-shelf COTS - based software systems.
- Highsmith, J. (2002). *Agile Software Development Ecosystems*. Addison Wesley.
- Jeffries, R. E. (2014). XProgramming.com An agile Software Development Resource. Recuperado el 17/7/14, <http://xprogramming.com/index.php>
- Pérez Pérez, M. J., & Gonzalez Cabrera, F. J. (s.f.). Guía comparativa de Metodologías ágiles. Universidad de Valladolid.
- Presario TI Latinoamérica. (Mayo de 2011). Presario TI Latinoamerica. Recuperado 18/3/13 http://www.presariotila.com/pdf/Especial%20E_RP_0511.pdf
- Proyectos ágiles ORG. (2014). Proyectos agiles.org. Recuperado el 10/7/14, de <http://www.proyectosagiles.org/>
- Riehle, D. (2000). A comparison of the value systems of adaptative software development an extreme programing: How methodologies may learn from each other. Recuperado el 31/7/14, <http://mailserver.di.unipi.it/estate/Library/XP/DOcuments/Papers/R2000.pdf>
- Rojas M., F. A., Delgado R., R., y Vera R., F. H. (2011). Diseño e implementación de un repositorio de componentes software para soportar el desarrollo de software empresarial – caso: División de Servicios de Información de la UIS. Iteckne, 8(2), 223-233.
- Schwaber, K., & Beedle, M. (2001). *Agile Software Development with Scrum*.
- Vera Rivera, F. H., & Rojas Morales, F. A. (2010). Propuesta de aplicación de la ingeniería del software basada en componentes en el desarrollo de software empresarial. Iteckne, 7(2), 128 - 135.