

Propuesta de metodología que coordine el levantamiento de requerimientos en apoyo a los estudiantes de la Universidad estatal de Sonora, unidad académica Navojoa, para introducirlos en el desarrollo global de software

Methodology proposal that coordinates the requirements gathering in support to the students of State University of Sonora, Navojoa academic unit, to introduce them in global software development

Josefina Ortega Ruiz

Universidad Estatal de Sonora, México
josefina-ortega@live.com.mx

Filiberto Valenzuela Mendoza

Universidad Estatal de Sonora, México
filiberto.valenzuela@ues.mx

Alma Isabel Arias Hurtado

Universidad Estatal de Sonora, México
alma_isabela@hotmail.com

Margarita Soto Rodríguez

Universidad Estatal de Sonora, México
maguysoto@hotmail.com

Resumen

El presente trabajo contextualiza el desarrollo global de software (DGS) como una nueva tendencia de trabajo, así como la posibilidad de crear una metodología para coordinar el levantamiento de requerimientos que apoye a los estudiantes de la carrera de Ingeniería en Software de la Universidad Estatal de Sonora, Unidad Académica Navojoa, a incorporarse en

esta área. Para ello, se abordan las problemáticas actuales en DGS y los beneficios de esta nueva tendencia de desarrollo de software.

Palabras clave: metodología para DGS.

Abstract

The present paper contextualizes Global Software Development (DGS) as a new trend of working, as well as the possibility of creating a methodology for coordinating the lifting of requirements to support students of engineering in Software of the University of Sonora, Navojoa Academic Unit, to be incorporated in this area. To do this, the current problems in DGS are addressed, as well as the benefits of this new trend in software development.

Key words: methodology for DGS.

Fecha recepción: Enero 2015

Fecha aceptación: Julio 2015

Introducción

En la actualidad, gracias a los avances tecnológicos es posible que las personas se comuniquen en un contexto laboral aun cuando se encuentren en regiones distintas. Algo similar sucede con el desarrollo de software: es posible la generación de proyectos con equipos, donde los integrantes se encuentren a kilómetros de distancia, lo cual es conocido como desarrollo global de software (DGS), el cual ofrece ventajas tales como la reducción de costos si se trabaja con regiones donde la mano de obra es más barata, por ejemplo, India y China. Asimismo, la diferencia de horario que existe entre diferentes países permite que el ciclo de desarrollo de software sea casi continuo y con ello pueda reducirse considerablemente el tiempo de término del proyecto. Siempre que se habla de las ventajas de un método de trabajo es conveniente investigar para descubrir sus desventajas y eliminarlas, mejorar la eficiencia de los equipos de DGS, maximizar los beneficios en esta nueva tendencia y, sobre todo, lograr que las nuevas

generaciones de estudiantes de Ingeniería en Software de la Universidad Estatal de Sonora se involucren en esta modalidad de empleo y logren vincularse con empresarios que están en la práctica bajo este entorno de desarrollo, o bien, que puedan crear entre los estudiantes equipos de desarrollo, bajo una metodología firme de levantamiento de requerimientos y trabajar de forma independiente dentro de esta modalidad de desarrollo de software.

El desarrollo de software es un marco de trabajo que se realiza con diferentes prácticas. Pressman (2008) menciona cinco prácticas que se deben llevar a cabo en todo proceso de ingeniería hablando de software: comunicación, planeación, modelado, construcción y despliegue. A continuación se describe la práctica de la comunicación.

El autor citado indica que “el camino desde la comunicación hasta el entendimiento suele estar lleno de baches”, lo cual significa que la comunicación efectiva es el mayor desafío para un ingeniero de software. En el libro de Pressman se detallan 10 principios importantes para que el levantamiento de requerimientos sea exitoso. Uno de ellos es “comunicación cara a cara”, ya que además de la comunicación verbal existe la gestual, la cual permite obtener más información acerca de los requerimientos del cliente. ¿Pero qué sucede en entornos de DGS? La comunicación cara a cara es nula o casi nula, a pesar de conocer que la fase de levantamiento de requerimientos es la más importante porque de ella depende el éxito o el fracaso de una aplicación. En la vida universitaria es prácticamente seguro que los alumnos no van a tener la suficiente experiencia para hacer un levantamiento de requerimientos, complicándose en un entorno global.

JUSTIFICACIÓN

La realidad actual del desarrollo de proyectos de sistemas de información es que se lleva a cabo con mayor frecuencia en un entorno distribuido, para lo cual se forman equipos de desarrollo en varios países, o las empresas conforman su equipo de desarrollo de software con personas en diferentes provincias dentro del mismo país; de esa manera ahorran costos y aumentan su competitividad. No obstante, la gestión para lograr estas ventajas en el entorno de trabajo es más compleja, ya que de no lograr una buena comunicación entre las personas que trabajan en lugares distintos, el producto final puede ser un caos, aumentando así los costos.

Algunos autores (Leandro Lopes, Rafael Prikładnicki, Jorge Audy), indican que el levantamiento de requerimientos es la fase más crítica en el desarrollo de software y que se complica aún más en entornos de DGS, ya que el marco ideal es que todo el equipo del proyecto (desarrolladores, clientes y usuarios) estén reunidos en el mismo lugar para facilitar la comunicación y llegar al entendimiento de los requerimientos de una manera favorable.

Al laborar en entornos de desarrollo distribuido, a menudo el equipo de trabajo consta de personas que hablan distintos idiomas, impidiendo la clara comunicación entre ellas y una comprensión correcta de lo que se desea lograr. Además, el aspecto cultural no debe dejarse fuera del ámbito de los requerimientos y de la coordinación de actividades entre los diferentes miembros del equipo de DGS. Esta manera de trabajar requiere de mayor cooperación por parte de las personas que provienen de diferentes culturas, pues existe diversidad en sus estilos de comunicación y sentido del tiempo, originándose serios malentendidos que afecten de manera negativa el desarrollo de los proyectos de software (James D. Herbsleb & Deependra Moitra).

Dichos aspectos que dificultan el trabajo en el entorno distribuido entre equipos de profesionistas, crecen cuando los estudiantes de nivel universitario desean poner en práctica sus conocimientos en el área e incorporarse a la modalidad de empleo, pero debido a su falta de experiencia se desmoralizan ante sus primeros intentos fallidos. Como recién egresados que se están incorporando a un ambiente laboral normal, donde los integrantes del equipo y los usuarios se encuentran cerca, en la misma oficina o ciudad, o bien con la posibilidad de sostener una conversación en persona, lo viven de manera más compleja incluso en entornos distribuidos.

Existen metodologías que apoyan el desarrollo de software, tales como RUP (Rational Unified Process), la cual nos da un enfoque para construir un software poniendo en práctica cuatro fases: inicio, elaboración, construcción y transición (Larman, 2006). En estas fases nos muestra lo que es necesario cubrir en cada una de ellas y los productos que debemos obtener, pero no nos indica cómo hacerlo.

Otra de las metodologías muy usadas actualmente son las metodologías ágiles, tales como la programación extrema (XP), SCRUM, el desarrollo adaptativo de software, entre otras, cuyo

principal objetivo es aplicar un conjunto de buenas prácticas para trabajar colaborativamente en equipo y de esa manera obtener buenos resultados en la realización de los proyectos, aunque no detallan específicamente cómo hacer un levantamiento de requerimientos, que es a lo que se enfoca esta investigación.

A la fecha, las metodologías antes mencionadas se imparten en las materias de Fundamentos de Ingeniería de Software e Ingeniería de Software en la UES, UAN, pero no es suficiente para ayudar al alumno a adherirse en el mundo del desarrollo global de software. Por ello es muy importante crear una metodología que apoye a los alumnos de la Universidad Estatal de Sonora, unidad académica Navojoa, a incorporarse a esta forma de trabajo, ya que esta prevalece cada vez más. No obstante, todavía existen muchos problemas que crean incertidumbre en el éxito de esta disciplina, mucho más tratándose de alumnos que carecen de experiencia laboral. Por esta razón, es muy importante aplicar métodos que incrementen su confiabilidad para continuar beneficiando con esta forma de trabajo.

A pesar del desafío, el DGS ha estado creciendo a un ritmo considerable, al grado de convertirse en una necesidad, pues aborda problemas tales como el costo, la escasez de recursos y la ubicación del desarrollo cerca de los clientes; asimismo, las problemáticas que limitan su óptimo desempeño deben reducirse.

PLANTEAMIENTO DEL PROBLEMA

Diversos autores han estudiado las ventajas y desventajas de la disciplina de desarrollo global de software (Oktaba, 2010-2011) (Eckstein, 2010). Entre las problemáticas que más mencionan están:

-La inexistencia de procesos exitosos que organicen la coordinación de un mismo proyecto entre varias organizaciones o equipos de trabajo.

-Problemas de comunicación: en la comunicación cara a cara entre equipo de trabajo y cliente son frecuentes los problemas de entendimiento, lo que se acentúa aún más con equipos trabajando en entornos geográficamente distribuidos, dando lugar a malentendidos, omisión de información, etcétera.

-Diferencias culturales: diferentes formas de pensar y de expresarse, diferentes costumbres y maneras de trabajar son una amenaza latente para la calidad del producto de software.

Estos problemas tienen como consecuencia re-trabajo, haciendo que la ventaja de costes deje de serlo, por la inversión extra de tiempo.

A pesar de dichos obstáculos, esta práctica es una realidad en la industria del software que va creciendo conforme pasa el tiempo. Con base en estas situaciones se plantea el siguiente cuestionamiento:

¿Qué impacto tendría el desarrollo de una metodología en los alumnos de la carrera de Ingeniería en Software de la Universidad Estatal de Sonora, Unidad Académica Navojoa, para que puedan coordinar el levantamiento de requerimientos de software, y después ingresar en el nuevo mercado laboral, como es el desarrollo global de software?

HIPÓTESIS

La metodología que se propone para coordinar de manera eficiente el levantamiento de requerimientos de software, ayuda a los alumnos de UES UAN a incorporarse en esta nueva modalidad de empleo de manera eficiente, de tal forma que se disminuye el problema de re-trabajo ocasionados por los malentendidos en las especificaciones dadas por el cliente, impactando positivamente en la reducción de costos de producción, tiempos de desarrollo y entrega de una funcionalidad completa del software, y generando la confianza de que los alumnos de la UES pueden ser considerados para trabajar en el medio.

OBJETIVO GENERAL

Desarrollar y evaluar una metodología para coordinar el levantamiento de requerimientos de software en un ambiente de desarrollo global distribuido, que apoye a los alumnos de la UES UAN a incorporarse en este enfoque de trabajo.

OBJETIVOS ESPECÍFICOS

- 1.- Aportar una herramienta metodológica para coordinar el levantamiento de requerimientos que ayude a reducir la problemática de re-trabajo en el DGS, impactando en los costos de producción y la entrega de software con funcionalidad completa al cliente y en los tiempos proyectados.
- 2.- Evaluar la metodología con equipos conformados por alumnos de la UES UAN.
- 3.- Comprobar a través de un estudio comparativo la disminución de los tiempos por re-trabajo en desarrollo de software, considerando sistemas proyectados en tiempos iguales, uno de ellos aplicando su forma tradicional para el levantamiento de requerimientos y otro utilizando el uso de la metodología desarrollada por la presente propuesta.
- 4.- Apoyar a los alumnos de la UES UAN a incorporarse al área laboral en un entorno de desarrollo global de software.

METAS

- 1.- Desarrollar una metodología para coordinar el levantamiento de requerimientos que guíe a los equipos participantes a plasmar de una manera clara las especificaciones del cliente y proyectarlas a los demás miembros del proyecto de una manera entendible, sin dar cabida a interpretaciones ambiguas.
- 2.- Aportar a través de la herramienta metodológica, un modelo a seguir para el levantamiento de requerimientos de proyectos de software en entornos de DGS.

MÉTODO

En la presente investigación se elaboraron diversas etapas que plantean una metodología que se adapta específicamente al levantamiento de requerimientos en un entorno de DDS. Dichas etapas son:

1. Iniciar una investigación minuciosa de las metodologías actuales orientadas al desarrollo de software, con el objetivo de asimilar los aspectos que abordan y rescatar la parte concerniente a la elicitación de requerimientos; posteriormente se analizan sus ventajas y desventajas para llegar a una conclusión del porqué todavía se tiene dicha problemática en esta área.
2. A continuación, llevar a cabo una entrevista con analistas de sistemas de diferentes fábricas de software que operan en modalidad de entorno distribuido, para recabar información importante de la operación real sobre los detalles de las problemáticas actuales relacionadas con el levantamiento de requerimientos.
3. Posteriormente se procede a desarrollar una metodología teniendo como fundamento la investigación de otras metodologías y la información recabada de expertos en el área, quienes han sido directamente afectados por la problemática que estudia la presente tesis.
4. Una vez que se tiene la metodología elaborada, seleccionar alumnos para un proyecto en la modalidad DGS, donde se pone a prueba la eficiencia de la metodología desarrollada; dicha selección se hace de manera minuciosa con el propósito de recabar resultados efectivos que ayuden a concluir si la metodología elaborada proporciona un apoyo importante en la elicitación de requerimientos, para que disminuya la problemática de la comprensión errónea en los requerimientos de software.

CONTRIBUCIONES ESPERADAS

A pesar de la complicación que representa el levantamiento de requerimientos en el proceso de desarrollo de software, no existe una metodología que contribuya específicamente a esta actividad; si se practica de manera errónea, se fracasará en el proyecto, incluso en los casos donde los interesados se comuniquen cara a cara. Dicho problema aumenta en entornos de DGS, antes mencionados. Sin embargo, esta forma de trabajo en el entorno global se ha convertido en una alternativa muy recurrente por parte de las organizaciones que tienen la necesidad de proveer

el desarrollo de software a más clientes, aprovechando el bajo costo de la mano de obra y la enorme cantidad de profesionales del desarrollo en algunas regiones (Damian, 2006).

Debido a lo anterior, es importante proporcionar una metodología eficaz que sirva de apoyo en la fase más crítica del desarrollo de software, es decir, el levantamiento de requerimientos, para que cada vez más los estudiantes tengan mejores bases en su vida universitaria y se adhieran sin mayores problemas a la vida laboral y, sobre todo, se incorporen a esta modalidad de trabajo en DGS, que de acuerdo con la investigación realizada tiene mucho que ofrecer a los nuevos profesionistas de desarrollo de software, sin importar dónde vivan.

Bibliografía

Damian, D. M. (2006). Global Software Development: How far have we come?

Eckstein, J. (2010). Agile Software Development with Distributed Teams. Dorset House Publishing.

James D. Herbsleb and Deependra Moitra, L. T. (s.f.). Global Software Development.

Larman, C. (2006). UML y Patrones. Pearson.

Leandro Lopes, Rafael Prikladnicki, Jorge Audy. (s.f.). Requirements specification in distributed software development a process proposal. Requirements Specification in Distributed Software Development a Process Proposal. Porto Alegre, RS, Brazil.

Oktaba, H. (2010-2011). Desarrollo global de software, retos y factores de éxito. Software Guru, 60.

Pressman, R. S. (2008). Ingeniería del Software. México, McGraw-Hill.