

Modelo de desarrollo esbelto de aplicaciones informáticas confiables**Modelo para aplicaciones confiables**

Ing. Kenny Orlando Suasti Alcívar. ⁽¹⁾
Ing. Sandy Raúl Chun Molina Mg.Sc. ⁽²⁾
Ing. Enrique Javier Macías Arias. ⁽³⁾
Ing. Nexar Bolívar Lucas Ostaiza. ⁽⁴⁾

⁽¹⁾ Instituto Tecnológico Superior Portoviejo, Ecuador
⁽²⁾ Instituto Tecnológico Superior Paulo Emilio Macías, Ecuador
⁽³⁾ Instituto Tecnológico Superior Portoviejo, Ecuador
Instituto Tecnológico Superior Paulo Emilio Macías, Ecuador
⁽⁴⁾ Universidad Estatal del Sur de Manabí, Ecuador

Contacto: Orlandokoka1980@outlook.com

Receptado: 02/10/2015

Aceptado: 14/12/2015

Resumen

El artículo presenta una propuesta de herramienta de manufactura esbelta aplicadas al desarrollo de software con seguridad lógica, dentro del marco de métodos ágiles, en particular el desarrollo de software seguro; lo que permitirá a equipos de desarrollo obtener una calidad sistémica del software (producto, procesos y personas que intervienen). Se recomienda emplear estas herramientas y evaluarlas constantemente con la aplicación iterativa e incremental del ciclo de calidad propuesto por Deming: planificar, hacer, verificar y actuar. Esta investigación se basó del campo tecnológico, documental, bibliográfico de carácter descriptivo, tipo encuesta la misma que fueron realizadas a diferentes profesionales en la materia, programadores y empresas dedicadas a la creación de software. Con esta propuesta se contribuye al desarrollo de proyectos de software con calidad en entornos científicos - académicos, ajustados al tiempo planificado y

Modelo para aplicaciones confiables

con los recursos presupuestados; utilizando herramientas propias de la ingeniería. Con la investigación se logra concluir que el desarrollo y la creación de software con seguridad lógica son más confiables y se benefician varias entidades públicas y privadas de nuestro país.

Palabras claves: Software seguro; Calidad de software; Método ágil; Ingeniería de software, calidad sistémica, seguridad lógica.

Slim model development of reliable applications

Abstract

The article presents a proposal for lean manufacturing tool applied to software development with logical security within the framework of agile methods, including secure software development; allowing development teams to obtain a systemic software quality (product, processes and people involved). We recommend using these tools and constantly evaluate them with the iterative and incremental implementation proposed by Deming quality cycle: plan, do, check and act. This research was based technological, documentary, bibliographic descriptive field, type the same survey were conducted with various professionals in the field, programmers and companies dedicated to creating software. This proposal contributes to the development of quality software projects in scientific environments - academic, adjusted planned budgeted time and resources; using proprietary engineering tools. The investigation concluded that manages the development and creation of logical security software are more reliable and several public and private entities of our country benefit.

Keywords: Secure software development, Quality software, Agile method, Software Engineering, systemic quality; logical security

Introducción

Las empresas de desarrollo de software no deben basarse únicamente en la experiencia de sus colaboradores para crear procesos y productos de calidad porque las aplicaciones informáticas cada vez son más complejas y exigen la colaboración de múltiples equipos. (Jose Ramón Rodríguez, 2007)

La industria del software, en el ámbito denominado sociedad de la información y el conocimiento, es considerada de gran importancia para la competencia en un mundo globalizado; por esta razón la calidad del desarrollo de software cobra vital importancia. (Calero, 2010)

Aunque la importancia de desarrollar software seguro ha sido ampliamente reconocida, la realidad es que en la práctica, la ingeniería del software no ha integrado debidamente la seguridad dentro del proceso completo de desarrollo.

Las relaciones entre las empresas, gobiernos y personas, ha cambiado por el uso de la tecnología; las empresas están interconectadas continuamente con los clientes y con otras empresas, además, utilizan herramientas de gestión y extracción del conocimiento para ser más eficientes; los gobiernos mejoran su presencia a través la Web y prestan servicios a los ciudadanos; las personas emplean redes sociales para sus relaciones interpersonales. La web 2.0 ha desarrollado una nueva sociedad virtual, donde el eje fundamental es la información; las computadoras y sistemas informáticos están omnipresentes en la sociedad moderna en lo que se denomina computación ubicua. (J, 2009)

Desde hace cierto tiempo surgió la rama de Ingeniería del Software, naciendo la inquietud de desarrollar productos de buena calidad, sin que esto signifique un incremento en el uso del tiempo, ni un mayor costo; se sabe que los sistemas informáticos son falibles al igual que la ciencia ya que son elaborados por seres humanos, lo importante es estar siempre en la búsqueda de perfeccionar las técnicas para evitar que los errores de software incidan en la operatividad de una organización o institución. (Vega, 2012)

Además, debido al uso generalizado y la confianza de las personas en los sistemas informáticos se hace necesario garantizar que cumplan con las expectativas de calidad y confiabilidad.

Este trabajo es la concepción y desarrollo de un modelo y plan de mejora que permita el desarrollo de aplicaciones informáticas con seguridad lógica, automatización de los procesos de análisis, verificación e implementación de protocolos de seguridad y, por lo tanto, sirva para evaluar la seguridad en software distribuido. Para la consecución de este objetivo final se han planteado los siguientes objetivos parciales:

Modelo para aplicaciones confiables

Rev. SINAPSIS, Vol. 7, N° 2, diciembre 2015

Análisis de intercambios específicos, funciones y mecanismos de seguridad actuales asociados a los protocolos de seguridad. Este estudio nos deberá determinar qué elementos esenciales habría que incorporar a las especificaciones formales con el fin de evaluar los ataques y amenazas.

Estudio y recomendaciones acerca de la implementación práctica de políticas de seguridad en sistemas reales. Este análisis permitirá tener conocimiento del contexto real en el que se desarrollan los protocolos de seguridad, así como las distintas posibilidades de implementación de servicios y mecanismos de seguridad en dispositivos físicos y niveles lógicos de las arquitecturas computacionales. (Sánchez, 2003)

Estudio de la evolución histórica de las técnicas formales en el campo de la seguridad, así como de las posibilidades de adaptación de las Técnicas de Descripción Formal actuales al análisis de la seguridad. El objetivo es incorporar a nuestro modelo una técnica de descripción formal que permita el análisis exhaustivo de ataques. (Jair, 2010)

Se revisaron los antecedentes que existen y se instrumenta, mediante herramientas específicas para un equipo de desarrollo de software en un entorno científico- académico, la aplicación de dicha filosofía, considerada un método de desarrollo ágil por diversos autores.

A continuación se presentan secciones donde se explica el desarrollo de software con seguridad lógica, los antecedentes de esta investigación, la calidad del software, las herramientas de desarrollo propuestas, finalizando con las conclusiones y recomendaciones.

Desarrollo de software esbelto confiable.

La empresa Toyota revoluciona en la década de los 80 a la industria automotriz con su sistema de producción que promueve la eliminación del desperdicio, resalta la cadena de valor del producto, manufactura bajo demanda (utilizando técnicas como la de justo a tiempo) y se enfoca en la gente que agrega valor. Womack, Jones y Roos son los primeros autores que adoptan el termino esbelto (*lean*) para referirse a esta filosofía de trabajo como un tipo de pensamiento denominado posteriormente *Lean Thinking*. (Hernandez, 2014)

La prioridad del sistema de producción de Toyota (TPS) es la velocidad entendida como: la

Modelo para aplicaciones confiables

capacidad de satisfacer la demanda del mercado con un sistema de producción en sintonía con la necesidad; y perfección: sólo siendo perfecto se puede ser rápido, sólo si no hay desperdicio (*muda* en japonés) se puede adquirir la velocidad necesaria.

Iniciativas esbeltas en manufactura, logística, servicios y desarrollo de productos han permitido mejoras dramáticas en costo, calidad y tiempo de entrega. Estos beneficios pueden obtenerse trasladando ciertas técnicas aplicables en el desarrollo de software. (Kioskea, 2012)

El pensamiento esbelto capitaliza la inteligencia de las personas que están más cerca de la agregación de valor al producto, ya que tienen la convicción de que son ellos quienes determinan y mejoran continuamente la manera de ejecutar su trabajo, tomando decisiones con autonomía; esto es a lo que se llama empoderar al equipo de trabajo.

Las técnicas de calidad aplicadas a la industria manufacturera no son trasladables automáticamente a la industria del software, sin embargo Poppendieck y Poppendieck han conceptualizado el desarrollo de software esbelto (*Lean Software Development*), el cual se basa en siete principios: eliminar el desperdicio, construir incrementalmente el desarrollo con calidad (mejoramiento continuo), preservar el conocimiento, diferir compromisos (tomar decisiones en el último momento responsable), entregas rápidas de funcionalidades (comprobadas), dar autonomía a las personas (empoderar el equipo de trabajo) y optimizar la visión completa (perspectiva sistémica del proyecto). (Sanders, s.f.)

Nociones tales como el desperdicio en manufactura se asocian a: generación de códigos inconclusos, sobre-documentación (papeleo excesivo), defectos (*bugs*), re-trabajos o asignar una persona a múltiples proyectos; evitar el transporte de productos es asemejado al cambio de actividades antes de finalizar funcionalidades; la sobre-producción es relacionada con la generación de características no requeridas por los usuarios finales.

Además se recomiendan la implementación de seis disciplinas: organizar el área de trabajo, establecer estándares (codificación, denominación, interfaz gráfica), control de versiones, procesos de construcción, integración frecuente y fijar políticas de pruebas (alcance, frecuencia, automatización y niveles de defectos).

Conjuntamente, las leyes de manufactura esbelta, según los citados autores, pueden aplicarse en el desarrollo de software en la siguiente forma: los clientes necesitan definir el nivel de calidad (pueden cambiar sus mentes), la velocidad es proporcional al tiempo desperdiciado en colas y esperas, el 20% de las actividades produce el 80% de los retrasos, la velocidad es inversamente proporcional a trabajos parcialmente elaborados (es preferible realizar implementación de conjuntos de características pequeñas en ciclos cortos de liberaciones de productos) y los costos de la complejidad son mayores que los de defectos o retrasos (no añadir características extras). (Schmid, 2002)

Asimismo afirman que la velocidad, calidad y el costo son características de un producto de software que están inexorablemente ligadas y todas pueden mejorarse sin ser afectadas mutuamente. Es por ello que la madurez en el desarrollo de software puede medirse por la velocidad en la cual un requerimiento del usuario puede ser trasladado al software repetidamente en forma confiable.

En términos generales se puede decir que el desarrollo de un software seguro logra expandir los basamentos teóricos del método ágil aplicando los principios bien conocidos y aceptados de manufactura esbelta a la elaboración de software.

Pero va más allá proporcionando herramientas de gestión para ayudar a trasladar los principios esbeltos

Materiales y métodos

Se utilizó en el proyecto, la investigación de campo, apoyándose en información originando encuestas, mediante este tipo de investigación se utiliza el método de análisis descriptivo, se logra caracterizar un objeto de estudio o una situación concreta, señalar sus características y propiedades. La recolección de datos se efectuó por medio de encuestas que fueron aplicadas a 50 profesionales que laboran en el desarrollo de software para diferentes empresas. La información se recopiló para realizar el diagnóstico del trabajo investigativo.

a las prácticas ágiles que son apropiadas para dominios individuales. (Garay, 2010)

Características.

Los 5 Principios que caracterizan al modelo ágil de Desarrollo de Software seguro.

1. Define el Valor desde el punto de vista del cliente: La mayoría de los clientes quieren comprar una solución, no un producto o servicio.
2. Identifica tu corriente de Valor: Eliminar desperdicios encontrando pasos que no agregan valor, algunos son inevitables y otros son eliminados inmediatamente.
3. Crea Flujo: Haz que todo el proceso fluya suave y directamente de un paso que agregue valor a otro, desde la materia prima hasta el consumidor.
4. Produzca el “Jale” del Cliente: Una vez hecho el flujo, serán capaces de producir por órdenes de los clientes en vez de producir basado en pronósticos de ventas a largo plazo.
5. Persiga la perfección: Una vez que una empresa consigue los primeros cuatro pasos, se vuelve claro para aquellos que están involucrados, que añadir eficiencia siempre es posible. (Kendall, 2005)

Beneficios

La implantación de Manufactura Esbelta es importante en diferentes áreas, ya que se emplean diferentes herramientas, por lo que beneficia a la empresa y sus empleados. (Sánchez M. R., 2010)

Algunos de los beneficios que genera son:

- ✓ Reducción de 50% en costos de producción
- ✓ Reducción de inventarios
- ✓ Reducción del tiempo de entrega (lead time)
- ✓ Mejor Calidad
- ✓ Menos mano de obra

- ✓ Mayor eficiencia de equipo
- ✓ Disminución de los desperdicios
- ✓ Sobreproducción
- ✓ Tiempo de espera (los retrasos)
- ✓ Transporte
- ✓ El proceso
- ✓ Inventarios
- ✓ Movimientos

Resultados

Las metodologías son un conjunto de mejores prácticas que si no se llevan a la práctica o se hacen a medias es muy difícil que se tenga calidad. El proceso de desarrollo de software es un proceso sociotecnológico. Se debe aplicar la metodología de desarrollo esbelto de software, si se tienen procesos bien definidos pero no funcionan de manera adecuada frente a los campos o bien, el equipo de desarrollo no está a gusto.

Las encuestas realizadas a los 50 profesionales de las diferentes empresas dan como resultado que el 68% de los software que existen en el mundo no son confiables y seguros, mientras que el 32% opinan que si son seguros. Se logra concluir que la mayoría de profesionales están de acuerdo que los recursos tecnológicos modernos ayudan a mejorar la calidad en la creación de software logrando llamar la atención de los propietarios de las empresas tanto públicas como privadas.

Los programadores profesionales manifiestan que el uso de recurso tecnológico moderno ayudará a fortalecer las seguridades del software y de esta forma no ser vulnerables.

Discusión

Incrementar la retroalimentación con el usuario, para poder determinar con la mayor exactitud posible lo que desea y resolver los problemas complicados que se presenten. La funcionalidad desarrollada debe ser exactamente lo que el usuario desea y que utilice (eliminando una fuente de desperdicio). (Sommerville, 2005)

Además se debe incrementar la retroalimentación dentro del equipo de trabajo, lograr que todos los involucrados conozcan el problema y participen en la resolución. Estimar el tiempo promedio que transcurre entre escribir la funcionalidad y las pruebas de desarrollo, las pruebas de cliente y su distribución. Luego, determinar un tiempo objetivo (a lo sumo 30% menos) por cada uno y atacar ésta lista de arriba hacia abajo para lograr el tiempo objetivo. Se recomienda para futuros proyectos, trabajar por iteraciones como ya se ha mencionado, tratando de cumplir con funcionalidades de los productos y no desarrollar con alcances muy grandes y complejos que no permitan las distribuciones rápidas. (Fernando Alonso Amo, 2005)

Tener un lugar destinado dentro del laboratorio del grupo de Base de Datos y programadores, así como en los laboratorios de los otros grupos involucrados un pizarrón donde los interesados en cada proyecto de desarrollo de software puedan ver: la meta de la iteración actual, ¿Qué se ha hecho?, ¿Qué se está haciendo?, ¿Qué no se ha hecho?, la misión del proyecto, qué se ha realizado para cumplirla y qué falta por hacer para cumplir dicha misión.

Conclusión.

Según lo analizado en la investigación realizada se puede concluir lo siguiente: Este trabajo ha descrito la naturaleza y características de la ingeniería de software en la seguridad de aplicaciones para el desarrollo esbelto de sistemas informáticos, la importancia de implementar una cultura de seguridad tecnológica profesional y cuál es el impacto que esta traerá a las organizaciones y sociedad en general.

Referencias Bibliográficas:

1. Calero, C. (2010). *Calidad del Producto y proceso Software*. Madrid: Ra-Ma.
2. Fernando Alonso Amo, L. M. (2005). *Modelo de desarrollo de programas*. Zaragoza, España: Javier Barbero Rubio.
3. Garay, S. G. (2010). *Integración Social Digital*. Mexico: Imagen Digital.
4. Hernandez, A. G. (2014). *El Gran Libro de Procesos Esbeltos*. Mexico: Ignius Media Innovation.
5. J, G. (2009). *Lean Software Development*. Argentina: UBA.
6. J., G. (2009). *Lean Software Development*. Argentina: UBA.

7. Jair, J. M. (2010). *Exploración de modelos y estandares de calidad para el producto software*. Ecuador: UIS.
8. Jose Ramón Rodríguez, J. G. (2007). *Gestion de proyectos informaticos, metodos, herramientas y casos*. Barcelona: UOC.
9. Kendall, J. E. (2005). *Análisis y Diseño de Sistemas*. Mexico: Pearson.
10. Kioskea. (2012). *Kioskea.Net*. Obtenido de <http://es.kioskea.net/contents/223>
11. Sánchez, J. S. (2003). *Ingenieria de Proyectos Informáticos*. España: Universitat Jaume I.
12. Sánchez, M. R. (2010). *Lean Manufacturing - La Evidencia de una Necesidad*. Madrid: Diaz de Santos Albasanz.
13. Sanders, J. (s.f.). <http://www.repositoriodigital.ipn.mx>. Obtenido de <http://www.repositoriodigital.ipn.mx/bitstream/handle/123456789/5321/34-2.pdf?sequence=2>
14. Schmid, s. R. (2002). *Manufactura, Ingenieria y tecnologia*. Mexico: Pearson .
15. Sommerville, I. (2005). *Ingenieria del Software*. Madrid: Pearson Educación, S.A.
16. Vega, V. Z. (2012). *Ánalisis comparativo de modelos de calidad*. Chile: Infonor.