APLICACIÓN DE INTEGRACIÓN CONTINUA PARA EL MEJORAMIENTO DE LA CALIDAD DE SOFTWARE EN SISFO LTDA.

YEIMY YULIANA YEPES LOPEZ

UNIVERSIDAD CATÓLICA POPULAR DEL RISARALDA

PROGRAMA DE INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES

PRÁCTICAS PROFESIONALES

PEREIRA

2010

APLICACIÓN DE INTEGRACIÓN CONTINUA PARA EL MEJORAMIENTO DE LA CALIDAD DE SOFTWARE EN SISFO LTDA.

YEIMY YULIANA YEPES LOPEZ

Informe de Práctica Profesional

Tutor

ALVARO IGNACIO MORALES GONZÁLEZ

Ingeniero de Sistemas

UNIVERSIDAD CATÓLICA POPULAR DEL RISARALDA

PROGRAMA DE INGENIERIA DE SISTEMAS Y TELECOMUNICACIONES

PRÁCTICAS PROFESIONALES PEREIRA

2010

CONTENIDO

Pág.

LISTA DE IMÁGENES

Pág.

LISTA DE ILUSTRACIONES

Pág.

LISTA DE APENDICES

RESUMEN

En el siguiente documento se muestra la manera de configurar un servidor que realice la integración de un proyecto de desarrollo, con el fin de mejorar la calidad del software dentro de la empresa, realizando pruebas automáticamente y estandarizando la forma de desarrollar software en equipo, esta práctica es conocida como Integración Continua.

Palabras clave: Integración continua, calidad de software, pruebas unitarias.

ABSTRACT

The following document shows how to set up a server to perform the integration of a development project, in order to improve the quality of the software within the company, through automatic testing and standardizing a practice to develop computer software, this methodology is known as Continuous Integration.

Key words: Continuous integration, quality of software, unit tests.

INTRODUCCIÓN

La calidad de software es una componente importante dentro de la empresa, es una preocupación a la que se dedican muchos esfuerzos con el fin de mejorar el producto proceso que generalmente no alcanza la perfección. El desarrollo de software de calidad, está acompañado por diferentes requisitos y normas que se centran en la implementación. Además de la certificación de los procesos para obtener un software de calidad, existen prácticas y herramientas que permiten alcanzar cierto tipo de mejoramiento en los desarrollos, una de las prácticas más usadas en los últimos años por grandes empresas de desarrollo de software, es la Integración Continua.

La integración continua permite que algunas tareas que son realizadas manualmente por los desarrolladores, como por ejemplo: pruebas unitarias, actualización y estandarización de código, detección de errores, etc. teniendo como consecuencia la automatización de tareas, la reducción de riesgos y mejoramiento de aplicativos.

El tamaño de la empresa y los procesos llevados a cabo son factores que se deben tener en cuenta al momento de aplicar la integración continua ya que se debe contar con diferentes entornos de desarrollo, prueba y producción para realizar una práctica adecuada debido a que no tiene sentido tener un servidor de integración continua y no tener un entorno de pruebas del desarrollo en donde se puedan revisar las aplicaciones antes de llevar a producción.

1. PRESENTACIÓN DE LA ORGANIZACIÓN

1.1 RESEÑA HISTÓRICA

SISFO, fue creada desde 07 DE FEBRERO DE 2007, sociedad de responsabilidad limitada, constituida inicialmente por 3 socios, ingenieros de sistemas, con un objeto social que comprende actividades principalmente encaminadas a desarrollar y a comercializar software. Sus inicios tuvieron como sede Parquesoft y luego un local en el centro comercial Fiducentro, actualmente está ubicada en el 9no piso del edificio de la Lotería del Risaralda, calle 19 No. 7-53.

La empresa ha tenido siempre como pilar fundamental, el desarrollo implantando nuevas tecnologías y lo último en plataformas, por lo que los socios propenden por estudiar y capacitarse cada día mas, tratando de replicar esto en los empleados, a quienes se apoya en cada uno de los proyectos de estudios y actualización que emprenden.

Los comienzos económicos de SISFO, fueron basados en el soporte a empresas en la adquisición e implementación y configuración de grandes servidores con las aplicaciones que estos requerían, en la actualidad se desarrollan grandes aplicaciones en la plataforma web, para empresas distribuidoras de productos y servicios, para quienes el soporte encontrado ha sido uno de las ventajas de la empresa.

El avance ha sido muy significativo en relación a los desarrollos generados, y a las empresas con las cuales cuenta como clientes son de gran dimensión en términos de su capital y su infraestructura.

1.2 DESCRIPCIÓN DE LA EMPRESA

Sisfo Ltda. es una compañía seria, moderna, con excelentes socios e ingenieros que permanecen siempre investigando, innovando y creciendo para entregar servicios tecnológicos de la más alta calidad.

Las especialidades de Sisfo Ltda. son: desarrollo de sistemas de información robustos y multiplataforma (aplicaciones para Internet), alojamiento web y dominios (Webhosting), soporte especializado para servidores Linux.

1.3 MISIÓN

Somos una empresa que brinda seguridad a sus clientes en el ámbito de la informática. Nos enfocamos en el desarrollo de aplicaciones multiplataforma y en el manejo de Servicios de Información, permitiendo a nuestros socios y colaboradores acceder a su información desde el sitio que quieran, usando cualquier tipo de dispositivo tecnológico actual.

1.4 VISIÓN

Consolidarnos como una empresa pionera en la prestación de servicios y creación de productos de óptima calidad, siempre avanzando a la par del progreso tecnológico. Aprovechamos la nueva era de las comunicaciones para globalizar nuestros productos y servicios.

1.5 POLÍTICA DE CALIDAD

En Sisfo Ltda. estamos comprometidos a:

- Ofrecer productos y servicios de alta calidad, con ejecuciones basadas en desarrollo ágil y gestión de proyectos garantizando tiempos efectivos de entrega y aplicación de estándares de desarrollo.

- Garantizar la satisfacción permanente de nuestros clientes, cumpliendo y superando sus expectativas.

- Mejorar continuamente los procesos identificados en la empresa, a través de la implementación de Sistemas de Gestión de Calidad.

- Poseer un equipo de trabajo altamente competitivo, comprometido con la compañía y que se encuentre en constante aprendizaje.

1.6 OBJETIVOS DE LA EMPRESA

- Crear productos y servicios de óptima calidad que sean reconocidos y recomendados por el gremio.

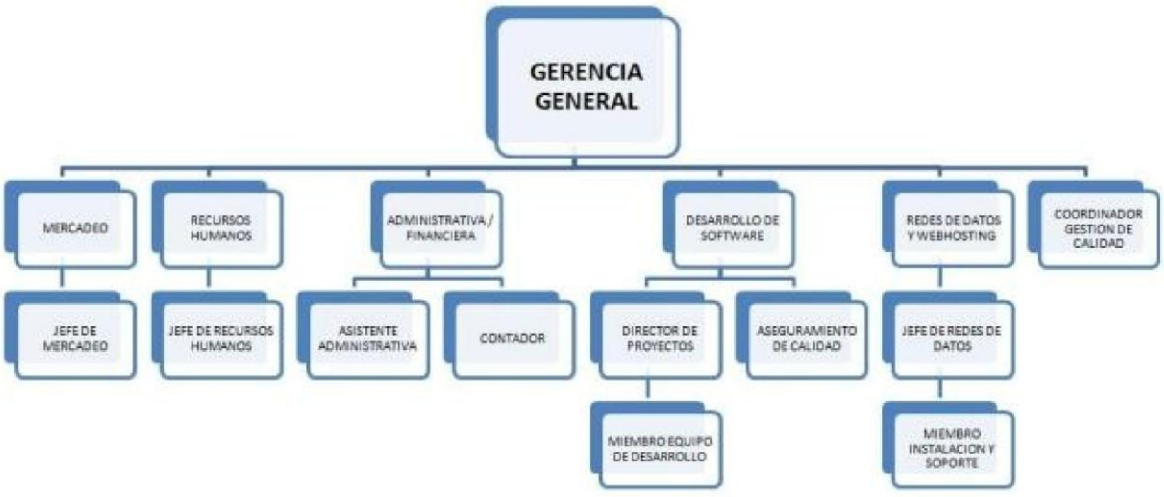
- Innovar en el mercado regional gracias a productos desarrollados con las últimas tecnologías y los más altos estándares de calidad.

- Trabajar en un ambiente amigable y honesto, que genere sentido de pertenencia y bienestar económico a socios y empleados.

- Ser reconocidos a nivel nacional como una de las empresas líderes en el desarrollo de software utilizando tecnologías modernas.

1.7 ORGANIGRAMA

Ilustración 1: Organigrama de la empresa



Fuente: Sisfo Ltda

1.8. ÁREAS DE LA EMPRESA

Área de desarrollo: Esta área es encargada de estudiar los procesos, metodologías y herramientas de producción de software, junto con los mecanismos disponibles para evaluar la calidad de los productos y la productividad de los procesos de desarrollo los sistemas de información.

La integración continua hace parte del área de desarrollo, se ubica aquí ya que es la herramienta unificada para realizar procesos automáticamente realizando test y actualizaciones y notificando a los desarrolladores cuando los procesos se hayan realizado exitosamente.

Área de redes: Es el área encargada de vigilar, organizar y soportar la red cableada existente en la organización basándose en las normas IEEE.

Área soporte: En esta área se encuentra el personal que está en la capacidad de prestar soporte a los clientes que cuentan con productos desarrollados por la empresa o que utilizan los servicios de alojamiento y hosting.

Área comercial: Esta área es la encargada de la comercialización del producto empresarial.

Área de diseño: Es el área encargada de realizar los diseños para los productos desarrollados con el fin de que cada paquete que se ofrezca, tenga un diseño único y creativo.

En total, la empresa cuenta actualmente con 8 personas.

2. DEFINICIÓN DE LA LINEA DE INTERVENCIÓN

La línea de intervención sobre la cual se desarrolla el proyecto es la de sistemas de información, ya que se pretende reunir diferentes elementos para ser orientados al tratamiento y administración de datos, en este caso de un repositorio de proyectos, los cuales deberán estar organizados y pasar por diferentes procesos automáticos de los cuales se encarga la integración continua**.**

La integración continua es la práctica de reunir el código desarrollado, verificar que funcione, con esto se ayuda a reducir errores dentro de la aplicación.

3. DIAGNOSTICO DEL AREA DE INTERVECIÓN O IDENTIFICACIÓN DE LA NECESIDAD

Teniendo en cuenta la realización de la encuesta (ver apéndice) se puede concluir que la integración continua aplicada en la empresa, puede solucionar muchas de las necesidades vistas en el área de desarrollo y que con la aplicación del proyecto de intervención, se logra reducir costos en términos de tiempo, mejorar la calidad de software y por ende, la calidad del servicio.

4. EJE DE INTERVENCIÓN

Siendo uno de los objetivos de la empresa crear servicios de óptima calidad que sean reconocidos y recomendados por el gremio es importante manejar procesos calificados que ayuden a cumplir con este objetivo.

Los procesos ágiles utilizados por la empresa para el desarrollo, permite que el desarrollo sea adaptativo haciendo que el código sea flexible y de fácil cambio.

El eje de intervención en este caso es la Integración Continua. Grandes empresas de desarrollo de software como Microsoft, IBM, Oracle, utilizan este proceso de integración continua diariamente en sus principales proyectos, teniendo como resultado clientes satisfechos.

5. JUSTIFICACIÓN DEL EJE DE INTERVENCIÓN

La empresa *Sisfo Ltda*. cuenta con proyectos que pasan por etapas de análisis, desarrollo y pruebas, utilizando un método ágil de desarrollo, siendo este una característica importante para la implementación de herramientas que ayuden a controlar la calidad del software.

Características como usabilidad, estándares, calidad, mantenibilidad, tiempo de respuesta en el desarrollo de software son objetivos importantes para lograr un producto completo y de calidad, pero la mayoría de veces no son muy tomadas en cuenta, para lograrlo se deben evitar ciertos tipos de prácticas que suelen pasar desapercibidas como por ejemplo: los malos hábitos de programación, aplicaciones no probadas previamente, aplicaciones difíciles y costosas de cambiar, aplicaciones con multitud de errores, código desactualizado, documentación inexistente o desactualizada.

Lo ideal es tener el código 100% documentado, conocer nuevas estrategias en construcción, monitoreo métricas de calidad , automatizar la documentación técnica, monitoria y el cubrimiento de las pruebas.

La integración de código en el trabajo de todo el equipo después de ciclos largos (semanas o meses) se torna una sesión de carga a defectos de integración. Integraciones poco frecuentes pueden generar atrasos en el cronograma y problemas de calidad en el producto. Sin integraciones frecuentes no es posible certificar diariamente la estabilidad del producto.

6. OBJETIVOS

6.1. OBJETIVO GENERAL

Hacer uso de diferentes herramientas y técnicas para iniciar el proceso de integración continua en Sisfo Ltda.

6.2. OBJETIVOS ESPECÍFICOS

Agrupar información clave en torno al tema de la integración continua.

Escoger las herramientas adecuadas para el proceso de integración continua.

Realizar la debida configuración en el servidor de integración continua para la automatización de tareas.

Orientar a los desarrolladores con el fin de que se creen test unitarios para todas las aplicaciones.

7. MARCO TEORICO

7.1 DEFINICIÓN DE INTEGRACIÓN CONTINUA

La integración continua es una metodología informática que consiste en hacer *integraciones automáticas* de un proyecto lo más a menudo posible para así poder detectar fallos cuanto antes. Entendiendo por integración la compilación y ejecución de test de todo un proyecto.

Según Martin Fowler, autor de libro “Continuous integration – Improving software quality and reducing risk”, el concepto de integración continua se define como sigue:

“Es una práctica de software donde los miembros del equipo de trabajo integran su código de manera frecuente, dando así múltiples integraciones por día. Donde cada integración forma parte de un Build (Integración, Construcción, Pruebas, Despliegue, entre otras cosas). „[1]

Hace algunos años el tema de calidad erar difícil de medir, aunque teóricamente se puede realizar revisando el código, pero en la práctica la inspección de código se hace pocas veces y es evidente que es necesario implementar metodologías para medir la calidad de software teniendo en cuenta aspectos imporantes en el desarrollo como el tiempo, las herramientas, las técnicas implementadas, etc.

Es por ello que encontrando los errores mas rápidamente reducimos costos y esfuerzo y esto se logra utilizando diferentes herramientas para medir la calidad, los estándares, tiempos de respuesta y trazabilidad de componentes partiendo de la necesidad de los requerimientos no funcionales que normalmente existen.

El concepto de Integración continua es una forma de desarrollo de software que mejora el proceso y puesta en producción del producto desarrollado, mediante la

[1] FOULER Martin. Continuous Integration – Improving software Quality and Reducing Risk. Estados Unidos. Pearson Education, 2007. p. 23

aplicación de herramientas, es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas web.

Se pretende implementar la integración continua, utilizando diferentes herramientas que permitan realizar la integración del código (al menos una vez por día) y adoptar un proceso que torne los resultados de la herramienta cruciales para avalar el estado del proyecto teniendo en cuenta que el desarrollo se realiza por medio de procesos agiles.

Un repositorio de datos, un servidor, una herramienta que permita automatizar la construcción de la aplicación y un IDE que se integre bien con el resto del entorno, son las herramientas base para lograr desarrollar la integración continua

7.2 PROCESO DE INTEGRACIÓN CONTINUA

El proceso paso a paso que se realiza al implementar un sistema de integración continua:

- Los desarrolladores del equipo hacen modificaciones en el código fuente, compilan y ejecutan las pruebas unitarias automatizadas y hacen el commit del código en la línea activa del desarrollo en la herramienta de control de versiones.

- La herramienta de integración continua de acuerdo a ciertos períodos de tiempo verifica si nuevo código se ha colocado en la línea activa del software de control de versiones.

- Si es así, la herramienta de integración continua extrae todo el código fuente y lo compila en el servidor que tiene por objetivo generar builds limpios en caso de que se deba hacer una compilación del código, lo cual no aplica en este caso debido a que los aplicativos de la empresa son orientados a un entorno Web.

- Después, la herramienta de integración continua ejecuta otras tareas que pueden ser definidas como ejecutar pruebas unitarias, pruebas de aceptación, generar información de las pruebas, de la cobertura y de análisis estático de código.

- La herramienta de Integración Continua actualiza los datos de la página Web del proyecto con todos los resultados de la ejecución del build con todas las fuentes alterados y con los datos de qué se hizo en cada iteración.

- "La herramienta Subversion, es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas web."[2] Esta herramienta es actualmente implementada dentro de la empresa por los desarrolladores lo cuales poseen los conceptos básicos para su utilización. Hoy en día es importante que al menos el código fuente del proyecto esté bajo un control de versiones ya que probablemente puede que no se tomen el proyecto seriamente y se pueda perder código o peor aún el proyecto completo.

La integración continua se hace importante desde el momento que exista más de un desarrollador trabajando sobre un proyecto o existan múltiples componentes en la aplicación los cuales agregan complejidad al proceso.

Para iniciar el proceso de integración continua, se inicia pasando una copia actualizada del código fuente a la máquina de desarrollo, esto se realiza por medio del software de control de versiones, que en este caso es el Subversion con el fin de tener una copia de trabajo a partir del código principal.

El software de control de versiones, mantiene todo el código fuente de un proyecto, en un repositorio, en cualquier momento un desarrollador puede hacer una copia de la versión principal a su máquina, proceso conocido como Checkout, esta copia ya ubicada en la máquina del desarrollador es conocida como working copy.

Después que el desarrollador completa las tareas en el working copy, se pasará el código a producción adicionando los test automatizados. La integración continua asume un alto nivel de pruebas que son automáticas en el software (código auto-testeable).

---

[2] FOGEL Karl. Producir Software de Código Abierto: Como llevar a un buen puerto un proyecto de código libre. Estados Unidos: O' Reilly Media, 2005. p. 304

Después de terminado el proceso anterior, en caso de ser necesaria la compilación de código, se procede a hacer un build ya automatizado en la máquina de desarrollo, en este caso no se realiza este proceso, ya que las aplicaciones no requieren de compilación, entonces se procede al ejecutar las pruebas automáticas. Como otras personas puedan hacer cambios en la versión principal, entonces, antes de hacer un commit, se debe hacer un update de la copia de trabajo del desarrollador que va a realizar los cambios.

Cada commit debe actualizar el repositorio principal en la máquina de integración. Usando commits diariamente, se tienen las aplicaciones testeadas ,esto significa que el repositorio principal está en el estado óptimo.

En caso de que en la práctica no se logre el proceso exitosamente se deben analizar situaciones como:

- Los desarrolladores no están actualizando sus versiones ni desarrollando el código para las pruebas.

- Existen diferencias de ambientes de desarrollo entre las maquinas de los desarrolladores.

Es necesario considerar que las pruebas se realizan en el servidor de integración y que cada desarrollador que realiza commit, debe ser responsable por el mismo, el debe monitorear el repositorio principal para que pueda corregir cualquier problema que ocurra.

7.3. IMPLEMENTANDO INTEGRACIÓN CONTINUA EN EL SERVIDOR DE SISFO LTDA.

"PhpUnderControl es una aplicación complemento para la herramienta de integración continua CruiseControl, que integra algunas de las mejores herramientas de desarrollo de PHP"[3]. CruiseControl incluye diferentes plugins para

[3] PICHLER Manuel, PHP Under control – Continuous Integration for PHP. 2010 http://www.phpundercontrol.org Disponible en http://www.phpundercontrol.org

una variedad de controles, las tecnologías de construcción y sistemas de notificaciones, incluyendo correo electrónico y mensajería instantánea.

El CruiseControl proporciona un interfaz web con los detalles de los proyectos y versiones anteriores.Para la integración continua se hace uso de phpundercontrol en el cual se incluye el CruiseControl, este también viene con una herramienta de línea de comandos para realizar todas las modificaciones de la instalación existente de CruiseControl.

PHPUnit es la aplicación más popular para xUnit PHP que proporciona un marco para las pruebas de software automatizadas. Salvo la automatización de pruebas PHPUnit pura contiene un conjunto de características como la cobertura de código, detección de problemas y métricas de software.

A continuación se muestra la instalación del PhpUnderControl y phpunit en el servidor teniendo el sistema operativo y la distribución de la máquina: Linux Debian Lenny

Se debe agregar a la lista `/etc/apt/sources`

```
deb http://ftp.debian.org/debian/ lenny main non-free
deb-src http://ftp.debian.org/debian/ lenny main non-free
# Actualizar para incluir los paquetes de la lista
apt-get update
 # Instalar xdebug este es necesario para phpunit
pecl install xdebug
echo "zend_extension=/usr/lib/php5/20060613+lfs/xdebug.so" &gt;&gt; /etc/php5/cli/php.ini
 # Instalar phpunit y phpundercontrol via pear
pear upgrade --force pear
pear channel-discover pear.phpunit.de
pear channel-discover components.ez.no
pear install phpunit/phpunit
pear install --alldeps channel://components.ez.no/Graph
```

```
pear install --alldeps channel://pear.phpunit.de/phpundercontrol-0.5.0

 # Descargar y extaer cruisecontrol

apt-get install unzip wget

cd ~

wget http://freefr.dl.sourceforge.net/sourceforge/cruisecontrol/cruisecontrol-bin-2.8.3.zip

unzip cruisecontrol-bin-2.8.3.zip -d /opt

cd /opt

ln -s cruisecontrol-bin-2.8.3 cruisecontrol

 # Correr  phpundercontrol con cruisecontrol

phpuc install /opt/cruisecontrol

# testrun

cd /opt/cruisecontrol

./cruisecontrol.sh
```

El servidor en donde se realiza la integración del código, cuenta con Apache y la herramienta de control de versiones Subversion ya instalados, estas herramientas son necesarias para realizar todo el proceso.

Los desarrolladores tienen un mismo IDE (Netbeans) para el desarrollo de las aplicaciones y cuentan con plugins de pruebas como lo es el Firebug, Web Developer y Css Validator.

7.4 VENTAJAS DE LA INTEGRACIÓN CONTINUA

Aumentando las implementaciones automáticas los procesos se realizan más rápido debido a que se logra reducir errores. El mayor beneficio de la integración continua es la reducción de riesgos, muchos desarrolladores piensan que están finalizando un proyecto de software sin saber realmente cuan largo debería ser antes de haber terminado. Es decir, un proyecto no termina cuando finaliza el desarrollo, después de esto se deben tener en cuenta las pruebas a las cuales debe estar sometido y un mantenimiento que puede variar en el tiempo.

Es muy difícil saber cuánto tiempo se tendrá en un proceso, el resultado es que los desarrolladores se ubican en un punto ciego, la integración continua reduce completamente este problema. No existe una integración larga y se elimina completamente el punto ciego, se sabe todo el tiempo en que parte del proceso esta, lo que funciona, lo que no funciona y los bugs pendientes que se tiene el sistema.

Los bugs suelen ser esos problemas que destrozan la confianza, atrasan los cronogramas y dañan la reputación. La integración continua no elimina los bugs, pero los vuelve dramáticamente mas fáciles de encontrar y remover, todo esto gracias al código auto-testeable.

Los bugs también suelen ser acumulativos, mientras más bugs mas difícil será el proceso de eliminación. En parte, esto pasa porque existen interacciones entre los bugs, donde las fallas son presentadas como el resultado de múltiples errores, volviendo cada error mas y mas difícil de encontrar.

Para el proceso es necesario la implementación de herramientas que se adapten al lenguaje de programación y la metodología usada por los desarrolladores, también es necesario la estandarización de IDE's que se utilizan así como otras herramientas para la detección de errores.

8. DEFINICIÓN OPERACIONAL DE TÉRMINOS

Commit: Es la operación que permite la validación de las actualizaciones del código fuente existente en el directorio de trabajo local de la maquina del desarrollador por medio de la herramienta de gestión de configuración.

IDE: (Integrated Development Enviroment) Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, compilador entre otras herramientas que ayudan al desarrollador en su tarea.

Subversion: es una aplicación para el control de versiones que nos permite gestionar los cambios y versiones que realizamos en nuestros desarrollos de una forma sencilla.

Test unitarios: es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.

Bug: Es el resultado de un fallo o deficiencia durante el proceso de creación de programas de ordenador o computadora (software). Dicho fallo puede presentarse en cualquiera de las etapas del ciclo de vida del software aunque los más evidentes se dan en la etapa de desarrollo y programación. Los errores pueden suceder en cualquier etapa de la creación de software.

Control de versiones: Es una combinación de tecnologías y prácticas para seguir y controlar los cambios realizados en los ficheros del proyecto, en particular en el código fuente, en la documentación y en las páginas web. Si nunca antes se ha utilizado un control de versiones, lo primero que hay que hacer es conseguir a alguien que sí lo haya hecho y hacer que se una al proyecto. Hoy en día todo el mundo espera que al menos el código fuente del proyecto esté bajo un control de versiones y probablemente no se tomen el proyecto seriamente si no se utiliza este sistema con un mínimo de competencia.

Build: Es el total de etapas necesarias para la compilación, creación de entregables al momento de ejecutar los test (funcional, unitarios, etc).

Procesos Agiles: Conocidos anteriormente como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Checkout: es la operación de extracción de una versión de un proyecto que se está desarrollando del referencial del administrador de configuración en un directorio de trabajo local.

Update: es la operación que permite la actualización a partir del referencial de la herramienta de gestión de configuración del directorio local.

9. CRONOGRAMA DE ACTIVIDADES PLANEADAS

Ilustración 2: Cronograma de actividades

Actividades	Julio		Agosto				Septiembre				Octubre				Noviembre				Diciembre	
	3º	4º	1º	2º	3º	4º	1º	2º	3º	4º	1º	2º	3º	4º	1º	2º	3º	4º	1º	2º
Investigación																				
Configuracion del servidor																				
Capacitación																				
Modificación de aplicaciones																				
Pruebas																				
Puesta en produccion																				
Mejoras																				

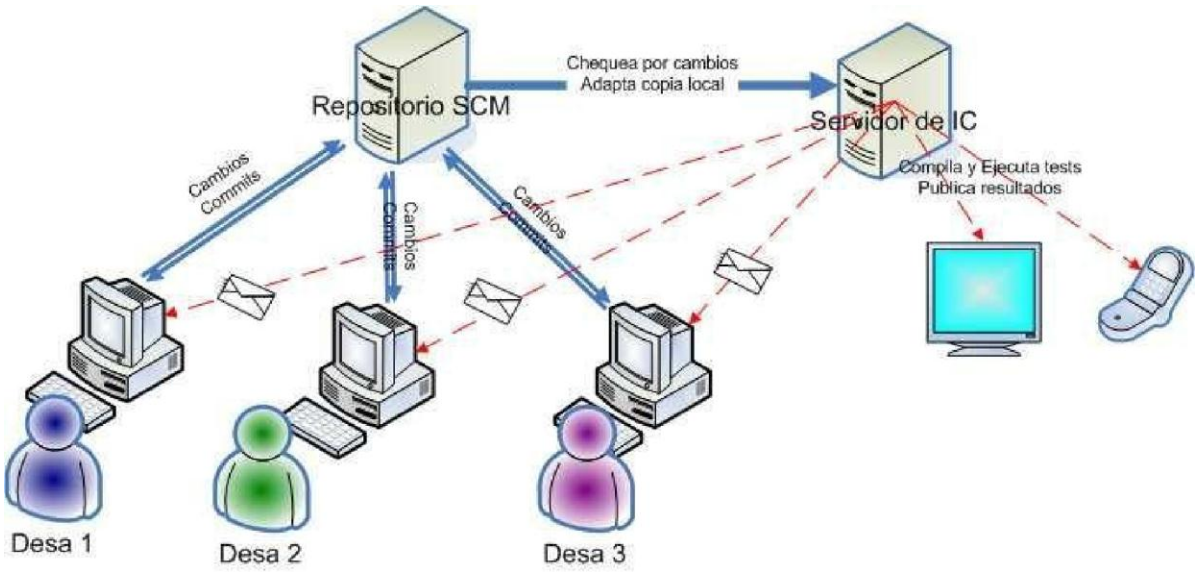
Fuente: Elaboración propia. Autor del informe

10. PRESENTACIÓN Y ANÁLISIS DE LOS RESULTADOS

El proceso de integración continua se logra agrupando herramientas y realizando una configuración adecuada en el servidor. La estructura final con la que se cuenta debe ser soportada por una persona al tanto de cualquier anomalía en el proceso.

El flujo de información ya aplicada la configuración en el servidor es el siguiente:

Imagen 1: Flujo de información en la integración continua



Fuente: IT Mentor http://www.it-mentor.com.ar/

En el gráfico se observa la forma de trabajo en donde:

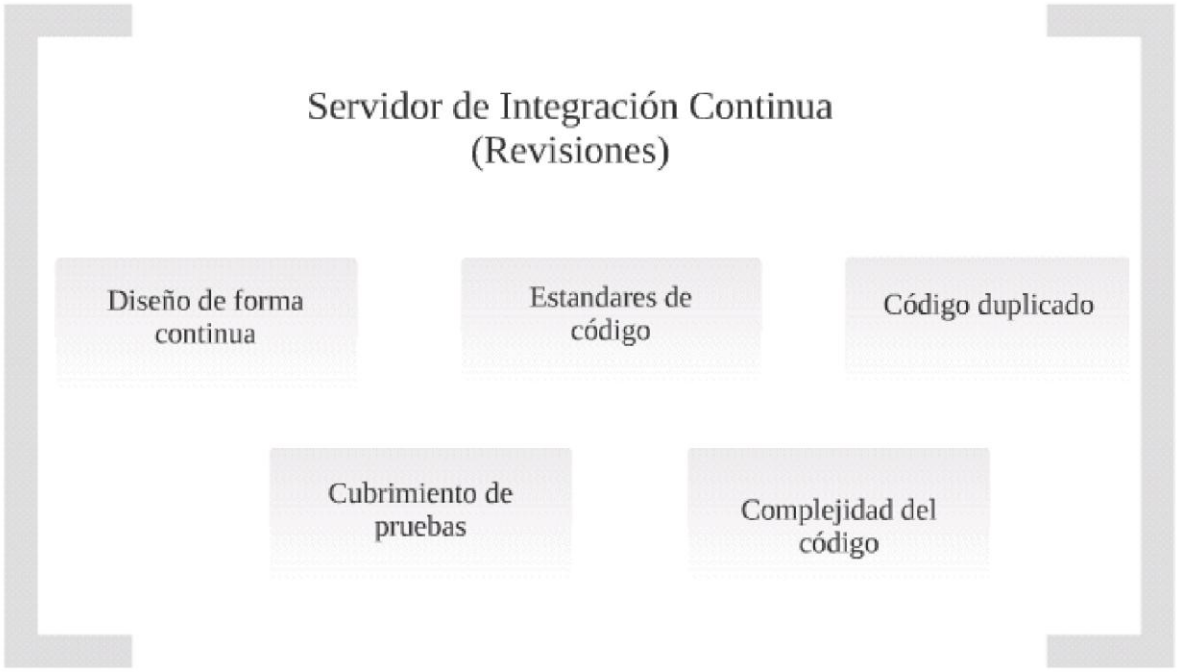
1. Un desarrollador realiza un commit (cambios) sobre el SCM (software configuration management) mientras el administrador de integración continua lo consulta por cambios con una frecuencia determinada.

2. Después del commit el administrador de integración continua detecta el cambio, toma del repositorio las últimas versiones y ejecuta los scripts que integran todo el software incluyendo cada una de las pruebas.

3. El administrador de integración continua informa por mail acerca de los resultados a los miembros del grupo de desarrollo de los resultados del build.

4. El administrador continúa consultando el repositorio con una frecuencia determinada.

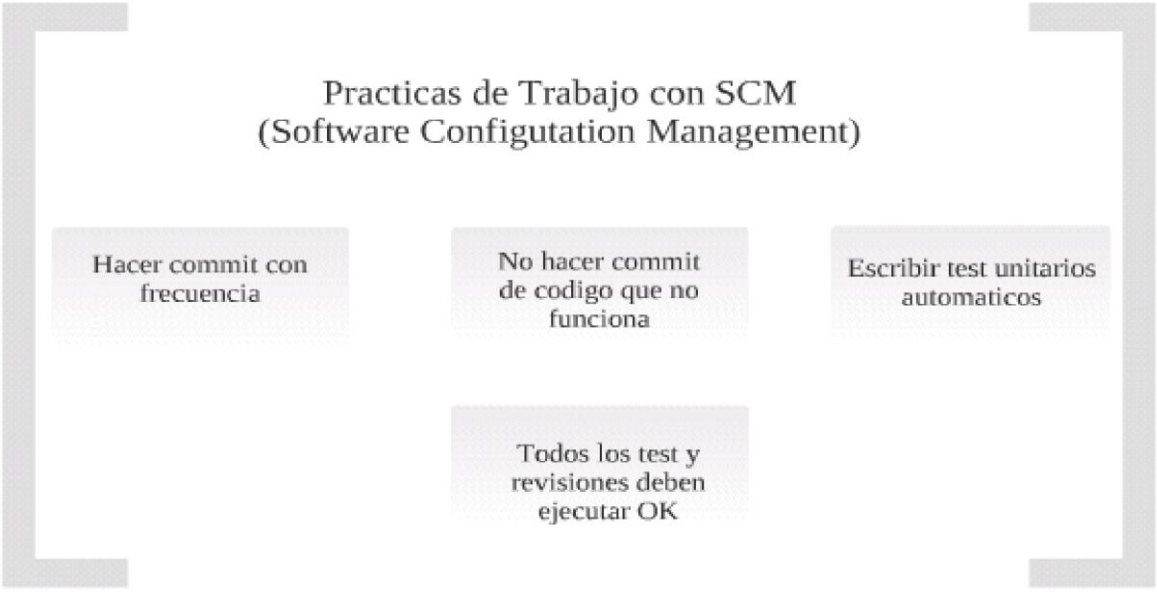
Imagen 2. Servidor de integración continua (Revisiones)



Fuente: Elaboración propia. Autor del informe

Para la correcta implementación de la integración continua, se deben tener en cuenta algunas prácticas de desarrollo.

Imagen 3: Prácticas de trabajo SCM



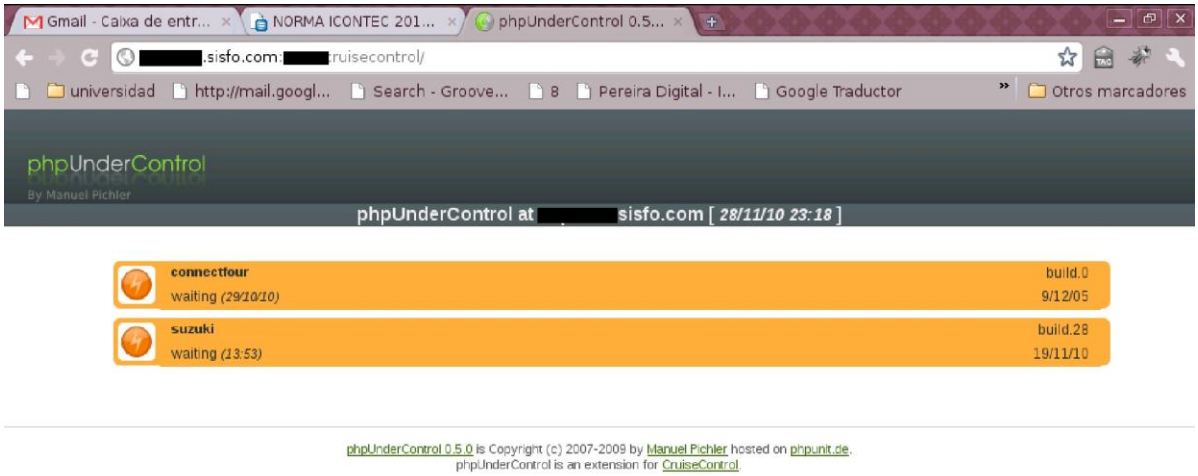
Fuente: Elaboración propia. Autor del informe

Los desarrolladores se encargan de realizar el código de los test previos a las aplicaciones; inicialmente lleva tiempo pero mientras se realicen mas test unitarios el cubrimiento de código será mayor y los test se realizarán en menor tiempo ahorrando el tiempo gastado en las pruebas clásicas de “ensayo y error”.

Automáticamente, el servidor de integración continua revisa los aspectos más importantes que hacen que un software sea catalogado como software de calidad, el servidor revisa los estándares de código en este caso PEAR, muestra el código repetido, el cubrimiento de las pruebas en porcentaje y el tipo de complejidad del código como por ejemplo la complejidad ciclomática que define como la medida cuantitativa de los ciclos diferentes que se encuentran en un fragmento de código.

A continuación se muestra la interfaz del entorno de integración continua bajo PHPunderControl.

Imagen 4: Pantallazo phpUnderControl



Fuente: Elaboración propia. Autor del informe

Pantallazo de interfaz de ingreso para ejecutar la integración continua de un proyecto.

CONCLUSIONES

El implementar integración continua dentro de una empresa de desarrollo de software, hace posible que los clientes obtengan un producto de calidad ya que este realiza funciones como la estandarización de código, pruebas, revisión de la estructura, documentación, entre otras aplicaciones que pueden ser añadidas con el tiempo facilitándole el trabajo a los desarrolladores debido a que estas tareas se realizan automáticamente.

La integración continua está totalmente relacionada con la comunicación, se debe tener claro que cada uno de los desarrolladores puede fácilmente ver el estado del sistema y los cambios que se han realizado. Para hacer integración continua, se necesita de múltiples ambientes, uno para correr las pruebas de los commits, otro para correr las pruebas secundarias, por lo tanto es importante tener scripts que permitan implementar la aplicación dentro de cualquier ambiente de manera ágil.

RECOMENDACIONES

Es importante tener en cuenta la configuración del servidor, la distribución del sistema operativo y las herramientas con las que se cuenta como lo es el IDE, o el entorno de pruebas, esto con el fin de evitar conflictos internos en la configuración.

Debe existir cierta cultura de desarrollo que permita trabajar en conjunto. Sin esos hábitos de desarrollo la Integración Continua no tiene ningún, valor debido al aumento de cambios y la complejidad de los errores que se pueden generar.

Todos los test unitarios deben ser correctos, no se puede decir que un software es de calidad si una parte de este falla en el momento de la implementación, por ello es importante que haya 100% de cobertura de código con los test.

Es necesario que los procesos realizados automáticamente, puedan ser visibles por los desarrolladores que participan, para así obtener mayor colaboración y conocer en qué etapa se encuentra un proyecto.

BIBLIOGRAFIA

FOULER Martin. Continuous Integration – Improving software Quality and Reducing Risk. Estados Unidos. Pearson Education, 2007. p. 23

FOGEL Karl. Producir Software de Código Abierto: Como llevar a un buen puerto un proyecto de código libre. Estados Unidos: O' Reilly Media, 2005. p. 304

PICHLER Manuel, PHP Under control – Continuous Integration for PHP. 2010 www.phpundercontrol.org/
Disponible en http://www.phpundercontrol.org

FOGEL Karl, Continuous Integration, 2006 www.martinfowler.com
Disponible en: http://martinfowler.com/articles/continuousIntegration.html

GALLARDO Pedro, Integración Continua, 2010. www.slideshare.com
Disponible en: http://www.slideshare.net/pedrogd74/integracion-continua

VELDMAN Mark, Getting started with phpUnderControl, 2009. www.techportail.ibuildings.com
Disponible en: http://techportal.ibuildings.com/2009/03/03/getting-started-with-phpundercontrol/

APENDICE A

ENTREVISTADO(A): Luz Adriana Betancur (Gerente)

1- Los productos desarrollados por la empresa están implementados para el uso interno?
R: Si, durante la implementación interna se realizan las pruebas antes de su lanzamiento al mercado.

2- Que áreas considera que necesita de mayor atención dentro de la empresa? **R:** Desarrollo y Soporte

3- Cual es el proceso de desarrollo que usted considera que necesita mayor apoyo?
R: Análisis y pruebas

4- Cree usted que las pruebas son importantes en todo el proceso de producción?
R: Muy importantes

5- Los clientes hacen parte de las pruebas de usabilidad del software?
R: Si.

6- Se evalúa el nivel de errores en las pruebas?
R: Si

7- Entre los desarrolladores evalúan sus propios desarrollos?
R: Si

8- Se realizan pruebas de integración?
R: Si